ALGORITHMS FOR MOLECULAR COMMUNICATION NETWORKS

A Dissertation
Presented to
The Academic Faculty

By

Bhuvana Krishnaswamy

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2018

ALGORITHMS FOR MOLECULAR COMMUNICATION NETWORKS

Approved by:

Dr. Raghupathy Sivakumar,
Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mary Ann Weitnauer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Matthieu Ratoslav Bloch
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ellen Zegura
School of Computer Science
Georgia Institute of Technology

Date Approved: June 18, 2018

ACKNOWLEDGEMENTS

I take this opportunity to thank each and everyone in my life who have helped me through this journey.

I would like to convey my sincere gratitude to my advisor Prof. Raghupathy Sivakumar. Siva has been a pillar of support and inspiration since my first day at Georgia Tech. He provided the perfect blend of guidance and freedom throughout my Ph.D. His passion and steadfast approach in identifying and understanding a problem are inspiring. At times of uncertainty, he helped me identify the questions that triggered my doubt and guided me to solve them. His insistence on understanding the fundamentals and being mindful of the impact and vision of a problem are some of the many lessons I take to my heart and would like to continue to ground my future research on these principles. My Ph.D. research with Siva has been a true learning experience that will shape my future endeavors. He is an inspirational researcher and a teacher that I was fortunate to work with.

I would like to thank my committee members Prof. Faramarz Fekri, Prof. Mary Ann Weitnauer, Prof. Matthieu Bloch, and Prof. Ellen Zegura for their valuable feedback and continuous encouragement. Prof. Fekri's feedback and discussions on the theoretical analysis provided a strong foundation for our results. Working with him, I learned many tips on proposal and paper writing. The first course I attended at Georgia Tech was Random Processes by Prof. Weitnauer. In spite of the class size, she interacted and helped students personally. She was always open to discussions and clarifications even outside the coursework. She has been instrumental in my job application and would like to convey my sincere thanks to her. I would like to thank Prof. Bloch for the many discussions during my Ph.D. that helped me shape my solutions. His diligence and the importance he gives to the understanding of fundamentals is a motivation for every student that work and interact with him. Prof. Zegura has been a mentor to me. Her door is always open to students who are interested in learning and brings out the best in each of us. Her continued encouragement

was invaluable to me during my job search.

I am fortunate to have worked on a collaborative research project with Prof. Brian Hammer, Prof. Craig Forest, Prof. Faramarz Fekri, and Prof. Ian Akyildiz. This dissertation would not have been possible without hours of discussion, brainstorming, and teaching from Prof. Hammer and Prof. Forest. Prof. Hammer helped me understand the bacterial system in consideration and the practical constraints to ensure we develop an impactful solution. Prof. Forest's enthusiasm, passion, and attention-to-details during our discussions and paper writing played a key role in the successful outcomes of our work. I want to thank Daniel, Caitlin, Patrick, Ozan, Arash and other students in the MONACO project who played a significant role in making this interdisciplinary work seamless. My sincerest thanks to Prof. Akyildiz for his mentorship, valuable feedback, and support. His encouragement and continuous support for my personal and professional growth are invaluable.

My thanks to Prof. P.V.Ramakrishna at CEG who introduced me to academic research and believed in me as a researcher. I want to extend my sincere gratitude to all my teachers at my undergrad, CEG, at my schools Sitadevi Garodia, Duthie School and Morton School. Each and every teacher helped me grow and inspire me in every walk of my life.

I wouldn't have enrolled in a research program, if not for my family. I am fortunate to be born in an inspiring and supportive family that considers education and learning a noble pursuit. I grew up listening to stories of my late grandfathers, Harihara Iyer and Bhagavatheeshwara Iyer, both teachers, their love for learning and teaching. My uncles Dr. Sivasubramaniam and Hariharan were the first people who got me interested in engineering. The love, support, constant guidance, appreciation, criticism, and feedback from my brother Dr. Hariharan and sister Meena is unmatched and closest to my heart. Most important of all, I would like to thank my mom and dad, who taught me from a very young age to ask questions and seek answers. I thank my nephews Vishnu, Hari, and Eshwar for bringing so much joy and excitement to our lives. I want to thank my husband, Dr. Jayaram Raghuram, one of the most important persons in my life. I feel fortunate for his support in

my personal and professional life. His curiosity and passion motivate to seek knowledge and get out of my comfort zone. I am thankful for his love, patience, and encouragement during my ups and downs.

I thoroughly enjoyed my 7 years at Georgia Tech and I owe it to all my amazing friends in and outside the campus that made Atlanta home away from home. My labmates Sandeep, Cheng-Lin, Shruti Sanadhya, Chao-Fang, Uma, Mohit, Yubing, Shruti Lall, and Nishit have been extremely patient to sit through multiple discussions and practice talks and provide feedback and criticism so that I was prepared. They made GNAN a fun place we all wanted to be. Sathya, Sneha, Ram, Ajay, Venkat and all the members of T-AMLI, Ravi, Sampath, Bala, Shilpa, Ranjini, Gautami, Rajiv, Harsha, Karthik, Swarnika, Nikita, Priti, Pragnya, and many many more who brightened my day even in moments of disappointment and rejections.

Last and most important, I thank God Almighty for this wonderful life and opportunity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Advancements in synthetic biology and bioengineering have allowed us to utilize naturally occurring, as well as engineered biosensors in sensing applications such as glucose monitors, blood haemostasis monitor [1], water quality monitor [2, 3], based on need and application. Biosensors have the advantages of abundance, natural occurrence, compatibility with their surroundings, sensitivity, and specificity to the bio-system. In most applications, biosensors are operated as independent units whose outputs are either processed off-line (lab tests) or are converted to electrical signals by means of transducers.

The goal of this work is to understand and develop fundamental communication algorithms to build an autonomous network of biosensors that can interact, cooperate, and respond akin to a natural system, which would create more opportunities for monitoring, diagnosis with a high level of accuracy.

Though the architecture of a bionetwork, consisting of biosensors and biological entities as transmitter and receiver, is similar to that of traditional networks (such as an EM network), the information transmitted, the system or devices used, and the channel/medium of information transfer are fundamentally different. This entails a rethinking of the design of communication algorithms for a bionetwork. We identify three fundamental communication problems that are essential to build a biosensor network and develop the following solutions to each of these problems identified. **Modulation** In a bacterial communication link, the processing delays at the transceivers and the propagation delays are of the order of a few minutes to a few hours depending on the environment. Hence, traditional modulation techniques are not directly applicable due to extremely high latency and low throughput. We developed Time Elapse Communication (TEC), a novel, practical, and non-linear modulation technique that encodes information in the time interval between the transmitted signals. TEC operates by sending a *start* and a *stop* counter that respectively triggers the receiver clock to begin and end counting. The number of clock cycles elapsed between

the two signals conveys the information. We also developed TEC-SMART to correct for the timing error in a non-ideal channel. **Addressing and Medium Access Control** In a star topology where multiple transmitters report to a single receiver, an addressing mechanism to identify and decode information from each transmitter is required. In traditional networks, the sender and receiver addresses are included as address fields. In a bacterial network, using additional bits for addressing would be inefficient and wasteful of resources. To address this problem, we designed Amplitude Division Multiple Access (ADMA), an embedded addressing mechanism wherein the transmitter address is embedded in the amplitude of the transmitted signal, eliminating the need for address fields. ADMA implicitly and efficiently solves the medium access control problem. **Reliability** To ensure accurate reception of signal amplitudes, we designed Amplitude-Width Error Correction (AWEC), a simple and efficient Forward Error Correction mechanism that can be implemented with biological circuits. AWEC introduces redundancy by varying the on-period of the signal transmitted and uniquely identifies each transmitter using the tuple $\langle$amplitude, on-period$\rangle$. While the number of unique amplitudes and on-periods available are limited by the system, AWEC effectively allocates and reuses them in order to assign each transmitter a unique id.

CHAPTER 1
INTRODUCTION AND BACKGROUND

Living organisms have evolved to monitor, sense, and respond to activities around them, acting as natural biosensors. Advancements in synthetic biology and bioengineering have allowed us to utilize these naturally occurring, as well as engineered biosensors, in sensing applications such as glucose monitors, water quality monitors, vital sign monitors, and more [2, 1]. Academic and industrial research programs are increasingly interested in the development of biosensors and the potential presented by this technology [4, 5, 6].

Currently, in the majority of applications, biosensors are operated as independent units whose outputs are either processed off-line (e.g., laboratory tests) or are converted to electrical signals by means of transducers. The former approach requires human involvement for post-processing which is time-consuming, expensive, and prone to errors. Electronic signal processing and communication devices, on the other hand, overcome these limitations, and also reduce the need for expensive and bulky devices for read-out (I/O device); for example, smartphone-enabled blood tests [7], wearable sensors to monitor health [8].

The latter approach, however, is limited by the existing devices, infrastructure, and by the requirement of compatibility of the communication network (wires or EM signals) with its surrounding system. Existing communication algorithms are not designed to overcome the challenges and/or exploit the opportunities that arise from a network with biological entities. For example, while energy efficiency is an important aspect in the design of algorithms for an EM network, the abundance and availability of biosensors that utilize natural sources of energy from their ecosystem provides opportunities for the designer to shift their focus on other aspects of the problem. In my doctoral research, we identify and explore the unique challenges involved in the design of a communication network of biosensors, and develop efficient algorithms and protocols for addressing these challenges.

Communication between biosensors can be categorized into two broad domains based on their target medium: *electromagnetic communication (EM)* extends traditional EM based communication techniques for use in non-biological applications [9, 10]; and *molecular communication (MC)* involves strategies for use in biological applications [11, 12, 13]. In recent years, bacteria have emerged as a promising candidate for nano-machines [14]. The nano-machines can be live biological nodes like bacteria [15, 16] or bio-inspired optical [17], and mechanical [18] systems. Recent advancements in synthetic biology and nanotechnology enable molecular communication possible and practical for applications like toxicology, environmental monitoring [3], and drug delivery [19].

The context for this work is molecular communication between bacterial populations. Specifically, we consider a system in which bacterial populations are used as transceivers connected through microfluidic pathways for molecular signals.

Bacteria are prokaryotic microorganisms, about 1 μm in size, that are well-studied and understood in terms of morphology, structure, behavior, and genetics. Genetic engineering of bacteria to introduce or delete DNA for specific traits (e.g., bioluminescence, motility) has enabled recent advancements in synthetic biology [20]. Many bacteria utilize a process called quorum sensing, whereby bacterial cells behave as transceivers that interact with each other, relaying signals by transmitting and receiving chemical signal molecules [21, 22].

Application Scenario

We consider a pathogen detection system, an example application of a molecular communication system with bacterial sensors to understand the network architecture, identify the challenges in implementing the system, and solve some of the key challenges to realize a molecular communication system. Since the topology and function of such a pathogen detection system is typical of other biosensing applications, we believe that the analysis and design of communication algorithms for this system is a significant step towards an autonomous bionetwork.

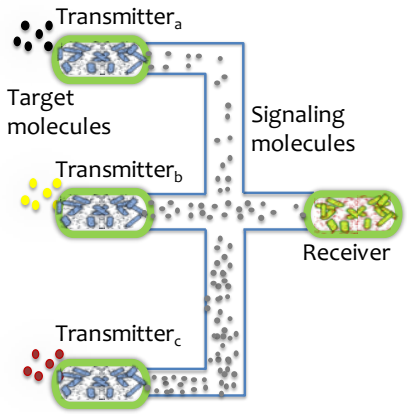


Figure 1.1: Pathogen Detection System

Genetically engineered bacteria have been developed to sense bacteriophages [23, 24, 25] and respond with a fluorescent protein. We present a pathogen detection application where a colony of genetically engineered *Sensor Bacteria* detect a specific pathogen in its vicinity and communicate this sensed information to a colony of *Receiver Bacteria* using chemical molecules. The sensor bacteria is a colony of genetically engineered bacteria designed to identify a particular pathogen and generate a carrier signal that conveys the information sensed. An illustration of the network architecture is presented in Figure 1.1. The sensor and receiver bacteria are housed in chambers in a microfluidic chip that has channels to carry the signaling molecules and nutrients to keep the colonies alive. The receiver, on receiving the carrier signal, fluoresce by generating green fluorescent protein. In the absence of a communication network, each sensor needs to be monitored individually, as each sensor fluoresces independently on detecting the pathogen. By building a network of these sensors and allowing them to communicate, multiple sensors can communicate to a receiver or sink; hence, only the sink is monitored continuously to read all the information. Such an architecture allows for more than one sensor to exist in the vicinity to detect a variety of pathogens or pathogens at different locations. Realizing such a network of bacterial sensors presents research challenges in the design and implementation of a variety of components as shown in Figure 1.2. We describe in detail each of these components

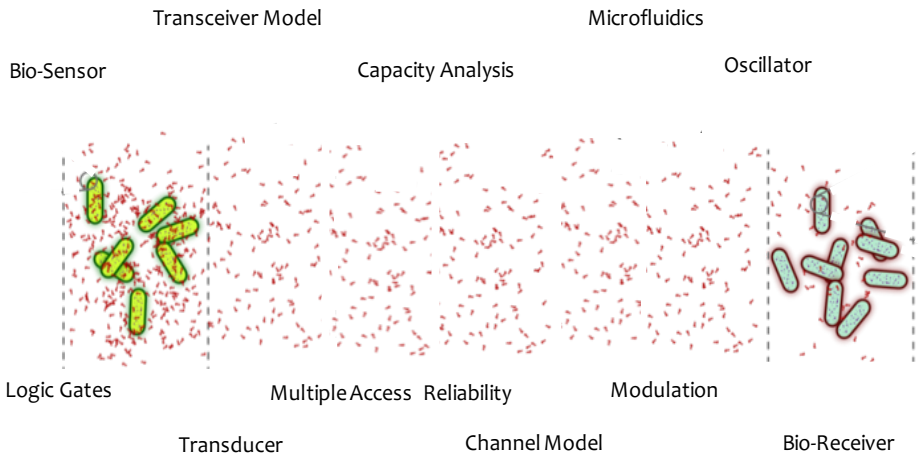


Figure 1.2: Bacterial Communication System Research

in Chapter [**2**]. In this work, we implement the following components to build proof-of-concept experiments, understand the system and develop communication algorithms.

- *Sensor:* The sensor is an engineered colony of bacteria, designed to detect a specific pathogen and output molecules to be communicated to a receiver.
- *Transmitter:* The transmitter takes input from the sensor, modulates and transmits the molecular signal into the channel.
- *Channel:* A medium of transmission between transmitter and receiver to communicate. A microfluidic chip that houses the bacterial colonies with a flow of nutrients serves as the channel.
- *Receiver:* The receiver is an engineered colony of bacteria that is designed to fluoresce on reception of molecules sent from the transmitter. The relative fluorescence at the receiver is proportional to the concentration of molecules received.
- *Optical read-out*: An optical readout to measure the relative fluorescence levels at the receiver, which is then used to decode the information transmitted.

Researchers have developed biological circuits bringing us closer to many of the components required to implement the pathogen detector. A detailed discussion of the survey

of past and ongoing work to realize many of these components is presented in Chapter [**2**]. Such a pathogen detection system can stop the onset of an epidemic by identifying disease-causing pathogens at an early stage. A successful implementation of the above architecture can open opportunities for further applications of autonomous biosensor networks.

We envision this detection system to be deployed on a surface in hospitals and household environments that require real-time monitoring of pathogens. We identify three fundamental communication problems that must be addressed to realize a practical pathogen detection system. We develop communication algorithms that consider the unique challenges and leverages the opportunities provided by the system. A pathogen detection system must detect not only the presence, but also the intensity of pathogens in the environment. To communicate the intensity to the receiver, a modulation technique is required. Multiple sensors are deployed to detect different pathogens and communicate with the receiver via point-to-point channels. This requires an addressing scheme that uniquely identifies each source and a multiple access control mechanism that allows the sources to share the single receiver. In this work, we identify the three problems of modulation, addressing and medium access control, and reliability as fundamental to building the pathogen detection system.

The first part of this work focuses on modulation techniques. Modulation is the process of varying the properties of a carrier signal with the information to be transmitted. On-Off-Keying (OOK) is a widely used modulation technique in MC research due to its simple design. But OOK is not throughput efficient in a high delay network. Thus, a modulation technique specific to slow networks, like Molecular Communication, is required. The majority of Molecular Communication research work focuses on models and algorithms for a single link with one transmitter and one receiver. As the number of transmitters and receivers increase, more challenges and opportunities arise. Some of them include : addressing, multiple access control, routing, reliability, and security. When multiple nodes access the same channel, without an addressing mechanism, we cannot identify the source

and/or destination. A mechanism to share the medium is also required. Following modulation, the next step towards building a communication system is to develop mechanisms for addressing; specifically, a local addressing mechanism, multiple access control, and reliability in the rest of this work.

To this end, we make the following major contributions:

- We use *Escherichia coli (E. coli)* bacteria genetically engineered to exhibit fluorescence upon the receipt of N-(3-Oxyhexanoyl)-L-homoserine lactone, or C6-HSL molecule. A microfluidic experimental system houses bacterial populations within micrometer-sized chambers fed by channels that provide both nutrients and controllable levels of C6-HSL, to demonstrate that a chemical signal at the sender can be reproduced as a fluorescence signal at the receiver reliably. Specifically, we demonstrate that it is indeed feasible to implement a simple modulation technique such as OOK for communication between the bacterial populations, but the consequent data rates achievable are as low as 10^{-5} bps. We define such environments where the transmission rates are very low as *super-slow networks*.

- We introduce a new communication strategy, *time-elapse communication* (TEC), for *super-slow networks* that rely on the time interval between two signals to encode information. Thus offloading some of the communication burden to the sender and receiver, we show that *TEC* under idealized conditions can deliver data-rate improvements of an order of magnitude. We also evaluate TEC under realistic conditions with non-zero error and show that the performance of TEC reduces to only marginally better than OOK. We propose an improved communication strategy, *smart time-elapse communication* (TEC-$SMART$), that improves data-rate performance in realistic non-zero timing error conditions. TEC-$SMART$ is a combination of two mechanisms viz., **Error Differentiation** and **Differential Coding**. The former decouples different components of timing error and corrects each component differently, and the latter reduces total delay by transmitting the difference of the

adjacent messages.

- We derive the maximum achievable capacity using TEC. We present an analysis of capacity for a uniformly distributed noise and an exponentially distributed noise in the microfluidic channel. Using simulations driven by experimental data, we also show that TEC-$SMART$ approaches the original promise of TEC even under realistic conditions involving non-zero error. We identify data-rate as a function of different parameters and perform a sensitivity analysis to analyze the impact of each parameter.

- We design an addressing mechanism for a topology with multiple sources and a single receiver. Such a topology is relevant in a sensing network, where multiple sensors report to or communicate with a single sink. As all sources communicate with the same receiver, an addressing mechanism to identify each source is required. We refer to such an addressing problem as *Source Addressing*. We developed *Amplitude-Division Multiple Access/Addressing (ADMA)* as a source addressing mechanism. ADMA uses the amplitude of the signal transmitted as the address of the respective source in the topology considered. Each source is assigned a unique amplitude. The sources transmit signals with the assigned amplitude. When multiple sources transmit simultaneously, the receiver receives the sum of amplitudes and identifies the components of the sum. Hence, the address of the sources implicitly solves multiple-access control.

- The number of sources that can be accommodated in the network using ADMA is limited by the maximum amplitude the receiver can receive. We developed an amplitude assignment algorithm that takes the number of sources and system constraints as input and outputs a sequence of amplitudes that maximize network throughput. We also design a computationally efficient decoder design that decodes the transmitted amplitudes given the received summation of amplitudes. We also study using

mathematical analysis, a theoretical upper bound on the throughput using ADMA. We propose to validate ADMA using real-time experiments and analyze the performance of ADMA using simulations in a NS3-based bacterial communication simulator, nanoNS3.

- ADMA embeds address in the amplitude of the signal transmitted. Accurate decoding of the amplitude is therefore required to identify the sender *and* the information. To ensure reliable reception of the amplitude carrying both the information and the address, an error correction mechanism is required. Due to the high latency of an MC system [26], feedback-based error correction mechanism will negatively impact the throughput performance and the complexity of system design. We propose Amplitude-Width Error Correction (AWEC) that introduces redundancy in the duty-cycle of the transmitted signal. We also design a decoder algorithm to detect and correct for amplitude error at the receiver.

- We identify the challenges in implementing a Forward Error Correction (FEC) in a bacterial communication system. We propose an embedded error correction mechanism that introduces redundancy in the duty-cycle of the transmitted signal. Each sender is assigned a unique 2-tuple id <**amplitude**,**on-period**>. The sender transmits bit 1 as a rectangular signal with an amplitude and an on-period uniquely assigned to the sender. The decoder observes the received signal and finds the closest 2-tuple id to the received parameters. We design a practical encoder and decoder to assign and decode the amplitude and on-period to maximize the decoding efficiency. We implement the proposed error correction mechanism in a Python-based custom-built MC simulator.

The rest of this dissertation is organized as follows: in Chapter [**2**] a survey of related work is presented. In Chapter [**3**], we present TEC, TEC-SMART and provide capacity analysis for TEC. In Chapter [**4**], we introduce ADMA, present the amplitude assignment

algorithm, decoder design, experimental validation and simulation results. In Chapter [**5**], we present AWEC, the decoder algorithm to detect and correct for amplitude errors and finally in Chapter [**6**], we conclude the work and present the challenges, directions of future research, and applications.

CHAPTER 2

LITERATURE SURVEY

As discussed in Chapter 1, a practical implementation of a molecular communication presents research challenges in the design of networking components, implementation, integration of these components, and communication algorithms. We identify the following components as the fundamental building blocks of a molecular communication network.

- Bio-sensor : The sensor that senses a biological signal or activity or process. Biosensors provide access to domains that remain inaccessible using other known sensing technologies. Advancements in synthetic biology have allowed us to utilize naturally occurring, as well as engineered biosensors such as water quality monitor [2, 3], detection of cancerous cells [25, 27, 24] among others. The accuracy, sensitivity, and delays involved varies with the biosensor and application.

- Transducer and/or Modulator : A transducer is a device that converts energy from one form to the other. Transducers are a part of independent sensor design. However, the design of transducer depend on the read-out mechanism, bio-receiver, and the environment, in a network of biosensors. The transducer also acts as a modulator that varies the characteristics of the information carrying signal from one sensor to the other.

- Processor : Processors form the core of a communication system. The fundamental building blocks of a digital circuit are NAND and NOR. Synthetically designed NAND and NOR gates have been developed using Bacteria [28, 29, 30] which paves way for a processor and transmitter using purely biological circuits.

- Channel/Medium :A channel or medium to carry the information from the transmitter to the receiver is required. Microfluidic chips and micro arrays have been developed

to house populations of bacteria [31, 32, 16]. Research focusing on designing different geometry for different applications, type of bacteria are presented in [16].

- Bioreceiver : A bioreceiver receives the information carrying signals from the transmitter and processes the received signal to decode the information. In molecular communication system, a bioreceiver consists of the transducer that converts bio signals to electrical signals, a demodulator to infer information from the signal and a decoder to decode the information and process for further use [33, 34, 35]. Majority of the existing biosensors use optical read-out. A read-out mechanism to process the received signal is also a component of the bioreceiver.

- Storage : A mechanism to store information before transmission is necessary to build a communication network. DNA storage, memory characteristics of microbial cells [36, 37, 38] indicates the progress towards biological storage.

- Oscillator : Clock is a key to run a processor to keep track of the information received, transmitted. [39, 40, 41, 14] Oscillators designed using bacterial colonies to achieve higher clock rates have been designed in laboratories and real-time implementations of oscillators is an on-going research.

- Modeling and capacity analysis: Accurate modeling of the channel and the transceivers is essential in understanding the medium and designing communication algorithms for the same. A number of works focusing on channel and system modeling [42, 43, 16], capacity derivation [44, 45, 46], modulation techniques [47, 48, 49] and analysis of channel and inter symbol interference [46, 50, 51] have been proposed. We provide a brief survey of some of these works in the following section.

- Modulation : Modulation is the process of varying the characteristics of a signal to convey information. The characteristics of the information carrying signal depends on the signal. Existing modulation techniques designed for electromagnetic carrier

signals must be redesigned for molecular signals. A survey of modulation techniques is presented in this chapter.

- Addressing and Medium Access Control : As we transition from a single link to a network, mechanisms to identify the transmitter, the receiver and medium access control mechanisms to allow multiple sensors to share the channel. Traditionally addressing is achieved with the help of address fields. MAC protocols have been developed for wired networks, wireless networks, and optical networks. Similarly, MAC protocol specific to biological networks that considers the delays involved, the channel being shared is required. We present a survey of MAC protocols and addressing mechanisms later in this chapter.

- Reliability : In order to ensure reliable transfer of information in a MC system, error correction mechanisms is needed. Specifically, low complexity error correction codes that can be implemented using biological circuits is an open challenge. The existing error correction codes are adapted from traditional coding techniques and do not consider the practicality of implementing these codes in a real-time, live bacterial system. The complexity of the code and its implementation is a crucial factor in the practical realization of the reliability mechanism.

- Routing: As the network grows and communication to a receiver is through multiple hops, routing algorithms will be required to find the best route to the destination. Based on the application and network topology, constraints in the design of a routing algorithm can change. High latency and low complexity make it challenging to design a feedback link, making network discovery difficult. Due to the dynamic nature of the transceivers and the channel, routing algorithms must be capable of adapting to node failures and route changes.

Following the above discussions, it can be noted that individual components required to implement show promise to be developed with the tools of synthetic biology. Inter-

operability of individual components is, however, a challenging task. The focus of this work is in the design of communication algorithms to efficiently convey information from the transmitter to the receiver.

Molecular communication research on modeling focuses mostly on channel modeling, but the design of receiver and transmitter play a significant role in the performance of the system. [42] develops an end-to-end model for molecular communication networks by modeling diffusion process using Fick's law, transmitter and receiver using ligand-receptor binding process. [43] using same models and derives the channel capacity of a molecular communication channel. The above models do not capture all the processes involved in the reception and processing of molecules by bacteria. In our work, we use a model developed in [16] that models population growth, transcription, translation, decay of molecules and microfluidic channel. The model is verified using experiments with genetically engineered *E. coli* bacteria in a microfluidic channel.

Modulation Schemes

On-Off Keying (OOK) is a simple modulation scheme widely used in Molecular Communication. OOK transmits a rectangular signal of a given amplitude and duration to transmit bit 1 ad no signal for a given duration to transmit bit 0.[44] derives capacity of channel and analyses the throughput achieved using OOK. The throughput performance of OOK is inversely proportional to average bit period. [47, 46, 48] proposes concentration shift keying (CSK) and molecule shift keying (MoSK) as two alternative modulation techniques that can improve the overall throughput. CSK encodes information in the concentration levels of transmitted molecular signal, similar to Amplitude Shift Keying. Higher the number of possible concentrations transmitted, higher is the throughput gain. Inter-symbol-interference (ISI) and channel noise affects CSK stronger that it affects OOK. MoSK is a modulation scheme in which, information is encoded in the type of molecule transmitted. Transmitter transmits different type of molecule for different messages. MoSK is not significantly af-

fected by ISI with the right choice of molecule types. But, the number of distinct molecules available to be used in MoSK is limited by the receiver system design. The complexity of receiver increases with increasing number of molecule types. [46] also proposes a molecular ratio based modulation where the ratio of concentration of two molecule type is used to encode information. [46] shows that the use of two molecules reduces complexity of system and ISI and impact of channel noise. [48, 52] also considers CSK and proposes frequency shift keying to improve throughput performance. Fine grained control over the frequency of molecular signal is difficult to achieve.

[53, 49, 50] considers timing based modulation techniques to encode information and derives capacity bounds on timing channels. [50] theoretically show that timing channel can be modelled as an inverse gaussian channel. Capacity bounds on the information in an inverse gaussian channel is derived. [53] encodes bits using timing channel. Bit 1 and 0 are encoded individually using time of release of molecules. Therefore, the throughput improvement over OOK is not possible whereas we encode entire message in the time between signals, thus improving throughput over OOK.[49] analyses the timing and energy based modulation combined. [49] proposes a modulation scheme in which the timing of a signal and the amplitude of signal conveys information to the receiver. They also derive bounds on the timing and payload combined modulation scheme in a molecular communication network. The above works do not consider a practical system and the timing error introduced by each component of the system. They do not provide a solution to detect or correct timing errors.

Addressing and Multiple Access Control

Majority of research on Molecular Communication focuses on channel and system modeling. Addressing and multiple access control are problems to be solved when more than one link exist. Most commonly used addressing is the use of address fields, such as MAC address and IP address. These addresses are unique to the device or the network card re-

sectively and can be used to uniqely identify the transmitter and the receiver. However, addressing overheads affect the throughput performance of high delays networks like bacterial communication networks. The following addressing mechanism have been proposed to reduce these overheads and achieve efficient addressing and medium access control. [54] proposes a distance based addressing mechanism. Based on beacons, propagation delay and path loss, transmitter estimates distance and follows chemotaxis like mechanism to reach receiver in a coordinate system. Such an addressing mechanism requires an accurate estimate of channel. Extending this scheme to more than one link is a challenge. A multiple access control protocol is needed to coordinate multiple transmitters accessing channel and receiver. Multiple Access Control is achieved using molecular type in [51]. Channel capacity of a multiple access channel assuming molecule type as a multiple access control mechanism is derived. Each transmitter communicates using a distinct molecule and hence do not interfere with other transmitters. But, the number of available distinct molecules is limited, thus affecting the scalability of the solution. Also, designing a receiver that can distinguish multiple molecules is a challenging problem. [55] propose an antenna/receptor design to improve reception efficiency and minimize demodulation and sampling error and [56] proposes decoding mechanism using analog filters. [44] analyzes the theoretical limits of information rate and [45, 16] propose mathematical models for the transceiver and channel. In this work, we propose to use signal characteristics to perform local addressing and multiple access control. The proposed addressing mechanism embeds address in the signal transmitted and implicitly solves multiple access. Thus, it does not add overheads or coordination between transmitters to perform multiple access control and improves network throughput efficiency.

Reliability

To ensure reliable reception of the amplitude conveying both the information and the address, an error correction mechanism is required. Due to the high latency of a bacterial

communication system [26], feedback based error correction mechanism will negatively impact the throughput performance of the network. Therefore, in MC system, forward error correction (FEC) mechanisms are suitable. FEC does not require any feedback from the receiver, eliminating communication overheads for error correction. FEC codes have been widely used in traditional communication systems. For example, cyclic redundancy checks for error detection, convolutional codes [**11ac**] to detect and correct bit errors, capacity approaching Low-Density-Parity-Check (LDPC) codes, Turbo codes to detect and correct bit errors without compromising throughput have been widely used in traditional networks. Existing FEC codes introduce redundancy by generating codewords, where individual bits are functions of the message bits; the redundant bits are generated by the encoder using matrix multiplication and/or polynomial multiplication.

Implementing these functions using biological circuits is highly challenging as the accuracy and consistency of the circuit design deteriorate with increasing complexity [57]. A number of research works have modified traditional FEC codes for MC without considering the practical constraints of an MC system [58, 59, 60]. [61] develops a family of ISI-free (Inter-Symbol Interference) codes that are simple and practical. The ISI free codes increase the Hamming distance between codewords and assign unique Hamming weight codewords to detect and correct codeword errors. Even though [61] provides a practical error correction code, it relies on a MAC protocol to handle collision. Any error caused by channel collisions will result in packet drop at the receiver.

CHAPTER 3

TEC :TIME-ELAPSE COMMUNICATION: IMPROVING DATA-RATES FOR MOLECULAR SIGNALING BY BACTERIA ON A MICROFLUIDIC CHIP

Microfluidic Experimental Results using OOK

In this work, we consider a system in which genetically engineered bacterial populations are used as transceivers connected through microfluidic pathways. Microfluidic pathways allow for dynamic changes in media composition. Further, the constant stream of media keeps the bacteria in ideal growth conditions, eliminating growth phase dependent variables from the experiments.

Genetically Engineered E. coli Bacteria

We establish an experimental system for testing the foundations of molecular communication in bacteria. To do this we utilized a marine symbiotic bacterium *Vibrio fischeri (V. fischeri)* which possesses a quorum sensing system called the LuxIR circuit. In standard laboratory conditions, the LuxIR circuit causes *V. fischeri* to generate light when a culture reaches an optical density 0.4 at 600 nM [62]. In the native system, the LuxI enzyme catalyzes the generation of a signaling molecule, C6-HSL. C6-HSL diffuses freely into and out of the bacterial cell. In the bacterial cell, C6-HSL binds with a second component, the LuxR receptor. LuxR, in complex with C6-HSL, binds specific DNA sequences and activates transcription of genes that are responsible for light production. In the native organism each individual cell serves as both transmitter and receiver. However, we ectopically expressed part of the LuxIR circuit in the model bacterial organism *E. coli* to engineer cells that only behave as receivers of signals. Specifically, we introduced into *E. coli* a plasmid that constitutively produced the LuxR receptor protein.

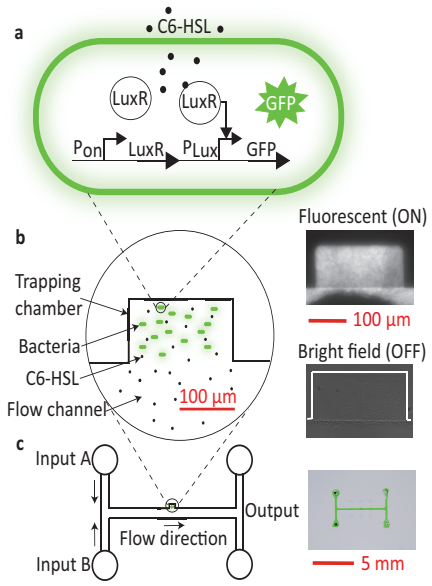


Figure 3.1: (a) Genetically Engineered *E. coli* Bacteria (b) Bacteria are housed in rectangular trapping chambers that are in fluidic contact to the main flow channel. As C6-HSL flows through the main channel, C6-HSL diffuses across the trapping chamber, which leads to the fluorescent response in the bacteria (fluorescent image inset). In the absence of C6-HSL, there is no fluorescence (bright field image).(c) Two inputs and two outputs are used in the microfluidic device adapted from Danino et al.[14]. (Photo of microfluidic device inset.)

Standard microbiological techniques were used in the culturing of *E. coli*. All experiments were performed in 2xYT broth [63]. *E. coli* strain DH5α was used for all cloning. Receiver bacteria were derived from the fully sequenced K-12 strain MG1655 [64]. To generate the receiver plasmid, Biobrick BBa_T9002 (partsregistry.org) was modified using PCR based methods to append a $ssrA$-degradation tag (ANDENYALAA) to the C-terminus of Green Fluorescent Protein (GFP) [65]. The resulting plasmid was transformed into MG1655 to create the receiver bacteria. The resulting strain exhibits fluorescence upon the receipt of a specic signal molecule C6-HSL, and is depicted schematically in Figure 3.1(a). When C6-HSL is added to the fluidic platform, it enters the receiver *E. coli* cells, LuxR complexes with C6-HSL and then binds to DNA sequences that induce transcription of an unstable variant of GFP (Figure 3.1(b)). A constitutive promoter (P_{on}) that is always on drives expression of the *luxR* gene that codes for the C6-HSL receptor, LuxR. When the C6-HSL signal reaches the receiver cells, it diffuses into the cell, and binds to LuxR. The

LuxR/C6-HSL complex activates the *lux* promoter (PLux), resulting in expression of the GFP gene carrying a degradation tag, and production of GFP. Engineered in this manner, receiver cells will become fluorescent in response to C6-HSL, and will stop being fluorescent when C6-HSL is no longer present.

Microfluidic System

Several other groups have examined responses of a bacteria to stimuli either in bulk culture or in a microfluidic environment [66]. In [67], the effect of population density on the ability of bacteria to respond was examined in microtiter plate wells. The effects of flow on receiver bacteria was examined in a microfluidic device in [68]. However, since poly-L-lysine was the method used to contain bacteria populations, experiments were limited to only to a few hours. Communication between two bacterial populations over time has been examined in [69] through means of a micro-ratchet structure and self-regulating populations that act as oscillators [70, 14]. Delivering a chemical stimulus in a time varying manner to a microfluidic bacteria while monitoring the fluorescent response was done previously by Groisman et al. [71]. In the current work, we advance this method by exploring the fundamental limits of pulse width. We modulate input signal using chemical cues to measure fundamental performance limits and ultimately to develop a new method of encoding molecular information surpassing these limits such that the data-rate is dramatically improved over OOK, the simplest form of amplitude shift keying wherein the presence of a signal (ON) represents a 1, and the absence (OFF) represents a 0.

Figure 3.1(c) shows an illustration of the microfluidic device. To fabricate it, we utilized standard soft lithography [72] with polydimethyl siloxane (PDMS) bonded to a glass coverslip. Briefly, PDMS (1:10) was cast on an $SU\text{-}8$ mold, plasma treaded with the grade 1.1 coverslip for 1.5 min, and bonded immediately following. During experiments, bacteria were maintained in chambers on the device (see Figure 3.1(b)) while bacterial growth medium (2xYT media containing ampicillin at 10 μg/ml) was delivered to flow channels alternatively with medium containing C6-HSL signal (note inlet A and B in Figure 3.1c).

The central flow channel (250 μm wide x 10 μm high) is in direct fluidic contact with the chamber (150 μm x 100 μm x 5 μm high) as shown in Figure 3.1(b),(c). In response to C6-HSL, the bacteria fluoresce (see Figure 3.1(b)), as imaged on a fluorescence microscope (Nikon TE 2000), with stage heated to (30°C). The microfluidic system included the microfluidic device on the microscope stage, pumps and tubing. To initially load bacteria on the chip, cells were injected in media through one of the inlet ports using a syringe to fill the chip entirely. Excess bacteria were flushed away, Tygon tubing was attached between the chip and pumps using short metal tubes, and the chip was placed on the microscope stage. The bacteria were then allowed to populate the chamber for 24 hrs until it reached capacity, $\backsim 10^5$ bacteria per chamber, during which time both inlets were used to flow 2xYT media at 100 μl/hr using syringe pumps (Harvard Apparatus). This flow rate was empirically determined to allow the bacteria to successfully colonize the chambers without being washed away.

Once the bacteria had filled the trapping chamber, combined flow rate was increased to 360 μl/hr. Inlet B was used for 2xYT medium alone (at 350 μl/hr), while inlet A (10 μl/hr) was used to varying concentrations and durations of C6-HSL as noted. Fluorescence images during the course of the experiment (1/10 min) were processed using MATLAB. For three consecutive images, a region of interest was selected that encompassed the chamber, the intensity of the pixels was averaged, and the background fluorescence subtracted out, yielding the signal strength. The obtained signal strength is defined as the relative fluorescence(y-axis) in Figure 3.2 and 3.3. The signal-to-noise (SNR) was then computed as the signal strength divided by the standard deviation of the background noise (non fluorescent bacteria-filled chamber).

Using the genetically engineered bacteria in the microfluidic system in Figure 3.1(c), we were able to elicit a fluorescent response to C6-HSL and image it with the fluorescence microscope (Figure 3.1(b)). At steady state (e.g., 1 hr) we were able to image fluorescent bacteria (number of experiments=10, SNR=20), and return them to non-fluorescing state

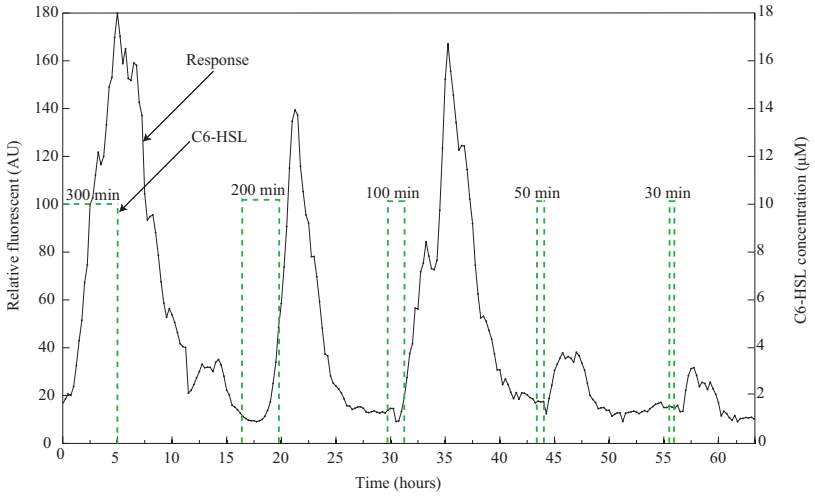


Figure 3.2: Bacteria relative fluorescence was measured in response to varying pulse inputs (300, 200, 100, 50 and 30 min) of C6-HSL. A typical response is shown.

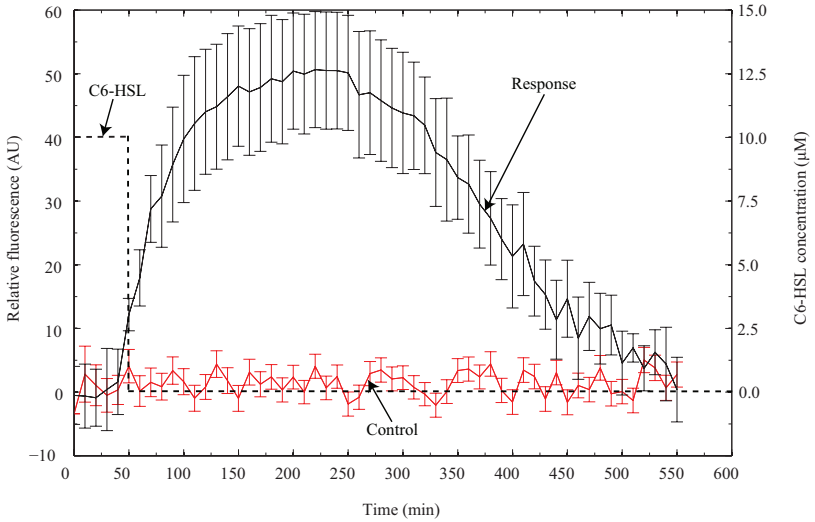


Figure 3.3: Response to 50min pulse duration

by removing C6-HSL from the flow channel (number of experiments=10, SNR$<$1). We experimented with modulating the C6-HSL input as a pulse with 10 μM concentration for a variety of durations. As shown in Figure 3.2, the bacteria responds differently to the varying input pulse with varying widths. In order to select an appropriate input pulse width, an experiment was run with varying pulses of 10 μM C6-HSL to determine the minimum pulse width that fit our requirements for a distinguishable signal. To be considered as a signal, we define a threshold signal-to-noise ratio (SNR) as $\geq$ 5, and a plateau region of sustained fluorescence above this SNR threshold of duration greater than 10% of the total signal time. Shown in Figure 3.2, the bacteria were exposed for 300, 200, 100, 50 and 30 mins with periods of pure media in between. The 50 min pulse was the shortest pulse that met these requirements, and was therefore used in the following experiments. The bacteria were exposed to C6-HSL for a 50 min pulse for all results shown in Figure 3.3. For ten samples, the average response time, defined as the time from when the bacteria begin to fluoresce until the time they stop, was found to be 435 min. with a standard deviation of 47. The average delay time, characterized as the time between when the bacteria start to receive the C6-HSL until they begin to fluoresce, was 31 min. with a standard deviation of 11. The average SNR was 7.9. We used the microfluidic system to demonstrate that OOK is (a) achievable in the target environment; and (b) has a data-rate performance that is quite low. *It can be seen that the receive signals clearly follow the ON-OFF patterns at the sending side, albeit offset by the propagation delay in the environment*. While the above results demonstrate that OOK can indeed be relied upon for conveying information from the sender to the receiver, we now derive the achievable data-rates using OOK based on parameters from experiments. The key parameter of interest in determining the achievable data-rate is the *bit period*. The bit period at the receiver is greater than that at the sending side due to the biological processing at the receiver bacteria. We define the maximum of the two bit periods as the effective bit period t_b. Acceptable SNR threshold used is an empirical value based on visual observation. The condition on SNR threshold determines

the effective bit period (t_b) of the system. Therefore, we analyze different values of t_b in our numerical analysis in Section 3.6. The data-rate of OOK is thus $\frac{1}{t_b}$, which for a t_b of 435 min is $3.8 * 10^{-5}$ bps. In the rest of the paper, we introduce and describe strategies that are aimed toward improving the achievable data-rates in *super-slow networks*.

Time-Elapse Communication

The data-rate performance of OOK in bacterial communication is low due to the inordinately large bit period involved. Hence, in this paper we explore a communication strategy called *time- elapse communication* (TEC), wherein information is encoded in the time period between two consecutive signals. A pictorial representation of TEC and OOK is presented in Figures 3.4,3.5. The number of molecular signals generated always remains at two (the start and the stop) irrespective of the number of bits required to represent the information. TEC requires the clock rates at the sender and receiver to be the same, although no clock synchronization is required. Intuitively, TEC improves the data-rate over OOK by reducing the number of communication signals that needs to be conveyed per unit of information. More precisely, if the clock rate at the sender and receiver is f_c, information v is represented by the sender as v/f_c time units separating a *start* signal and a *stop* signal, where $v \in N$. If the communication involves conveying a series of such values, the *stop* signal of a particular value is used as the *start* signal of the next, and hence the number of communication signals per unit of information is amortized to just one [1] In OOK, an information value v would be represented using approximately $\log_2 |S|$ bits, ($|S|$ is the cardinality of set S) where $v \in S$. A value of 5 is represented using 3 bits and requires $3t_b$ time units. However, in TEC, v is represented using v clock cycles, and hence the clock rate has to be exponentially larger than the underlying OOK data-rate in order for TEC to exhibit superior performance. Revisiting the set-up in Section 3.1, for an OOK data-rate of $3.8 * 10^{-5}$ bps and a clock rate of 1 Hz, under idealized channel conditions, TEC

[1] Clocks are prevalent in bacteria. Synthetic clocks have been generated and their rates altered by genetically manipulating *E. coli* bacteria to include genes from a variety of bacteria.[70].

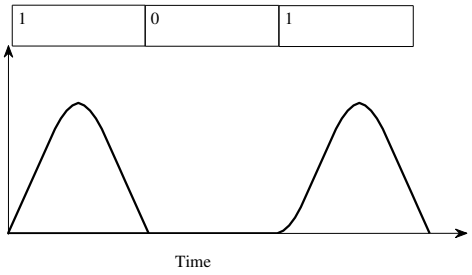


Figure 3.4: OOK

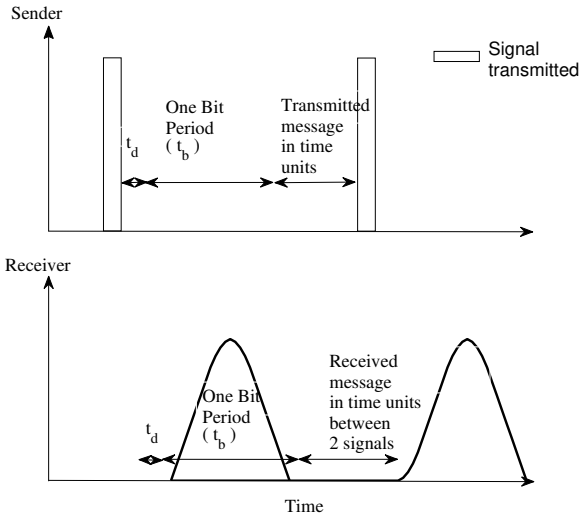


Figure 3.5: TEC

will provide an average data-rate of $3.9 * 10^{-4}$ bps, a 10.3x improvement over OOK. In general, consider a decimal value i being sent, the total delay required to communicate this data using TEC is the sum of one bit period using molecular signaling and the information delay (say $t_{in} = \frac{i}{f_c}$) corresponding to the wait time for the data. Thus, it takes TEC a maximum of $t_b + \frac{2^n-1}{f_c}$ time to transmit a n bit data. The data-rate of TEC is thus given by,

$$R_{tec} = \frac{n}{t_b + \frac{i}{f_c}} : i \in \{0, 1....2^n - 1\}. \tag{3.1}$$

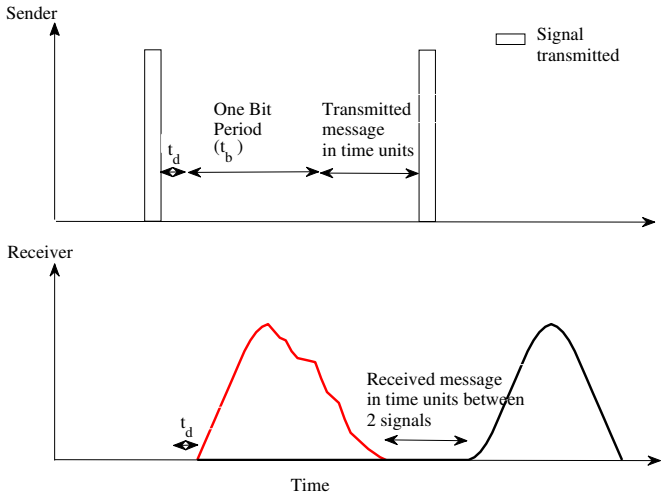
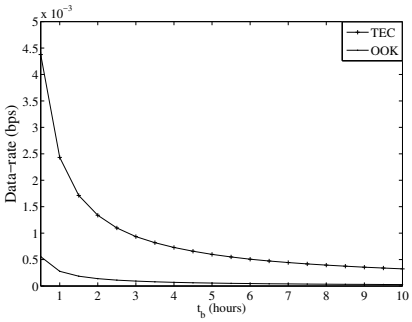


Figure 3.6: TEC - Non zero error

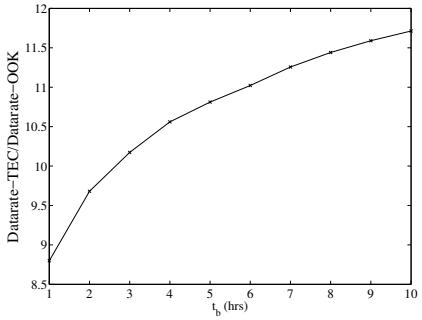
The notion of encoding information in time is not new to this work. *Timing channels* rely on such a notion to achieve covert information transfer [73], while *Pulse-Position Modulation (PPM)* relies on conveying information through the relative position of pulses. The key difference between such techniques and this work is significant: the domain of interest - bacterial communication - raises unique and considerable challenges in how a technique like TEC can be realized in the target environment.

Promise of TEC:

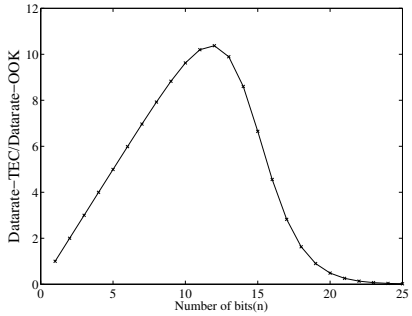
We now use numerical analysis of the data-rate equations of OOK and TEC to study the promise of TEC under variations of different parameters. Unless otherwise specified, we use a bit period t_b of 435 min based on the experimental results presented in Section 3.1, and a clock rate of 1 Hz. The data-rates of OOK and TEC as a function of t_b is shown in Figure 3.7a, while Figure 3.7b presents the relative performance improvement of TEC with respect to OOK. With an increasing t_b, TEC's improvement over OOK increases since the dependency of TEC's performance on the parameter is relatively smaller. Figure 3.7c presents the relative performance improvement of TEC with respect to the number of bits n. Thus, for a given t_b and f_c, there is an optimal value of n. Finally, if f_c is higher, the waiting time between signals will be smaller. It can be observed from Figure 3.7d that



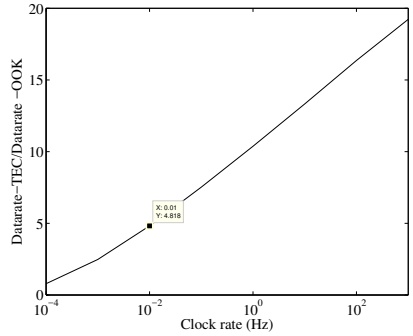
(a) Effect of Bit Period



(b) Bit Period



(c) Frame size



(d) Clock rate

Figure 3.7: Performance of TEC under ideal zero error conditions

TEC's relative performance with respect to OOK improves with higher f_c. Note that while a higher f_c is always better under zero error conditions, any skew in clock rates between the sender and the receiver will be exacerbated under realistic non-zero error conditions.

Limitations of TEC:

Thus far, we have explored the performance of TEC under idealized zero error conditions. In reality, the responses of biological systems varies across time. Figure 3.6 illustrates a deviation from ideal behavior. The start signal in Figure 3.6 gets delayed and hence the time elapsed between the signals is different leading to bit errors. To the best of our knowledge, there has not been any work that models the statistical distribution of the delay in the response of bacteria to molecular signals. Hence, we consider a simple uniform distribution $U\ (t_b\text{-}\epsilon, t_b\text{+}\epsilon)$ to model the real response time of receiver bacteria. On an average, one bit period is t_b with a bounded error that is uniformly distributed $U(\text{-}\epsilon, \text{+}\epsilon)$. Any deviation from the average is termed as error. The net error ϵ is the sum of all errors from the time of

introduction of molecules into the medium to the detection of fluorescence output. Given that the error is bounded, it is possible for the receiver to decode with 100% accuracy by increasing the minimum distance between messages. A message is defined by *start* and *stop* signals, and both these signals are subject to an error of $\pm\ \epsilon$. If the minimum distance between adjacent messages is at least 4ϵ, the receiver can decode messages correctly in spite of any errors. We refer to TEC with simple error correction as $TEC\text{-}SIMPLE$. Figure 3.12 shows that the relative data-rate performance of $TEC\text{-}SIMPLE$ in a realistic system has reduced to approximately 1.8x OOK (for an error of 10% in t_b). Thus, the introduction of error in the system has brought down the performance of TEC considerably.

TEC-SMART : TEC for Non-Zero Error Conditions

In this section we propose multiple techniques that in tandem improve the performance of TEC under non-zero error conditions. We refer to a communication strategy that uses TEC along with these techniques as *smart time-elapse communication* ($TEC\text{-}SMART$). Specifically, we present (i) an error curtailment/differentiation strategy that reduces the impact of error on $TEC's$ performance; (ii) a differential coding strategy that is uniquely targeted towards amortizing the cost of t_b across multiple pieces of information; (iii) an optimization to the differential coding strategy that reduces overheads and (iv) an optimization to detect error in case of unbounded channel noise.

Error Curtailment/Differentiation:

The timing error is the sum of multiple error components: propagation-time error e_d, rise-time error e_r, and fall-time error e_f corresponding to the propagation of molecules through the medium, the ramp-up of fluorescence, and the ramp-down of fluorescence respectively. Instead of handling the composite error in its entirety, we propose handling the error in two independent stages by introducing redundancy in the *bit period* to handle e_r and e_f, and in the *information delay* to handle e_d.

Fall-Time Error Correction:

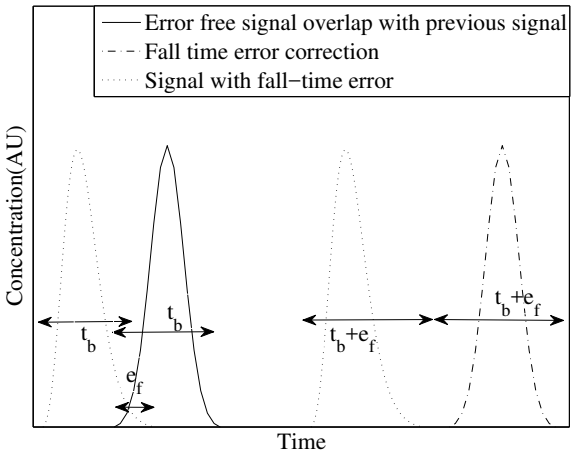


Figure 3.8: Illustration of Fall time error correction

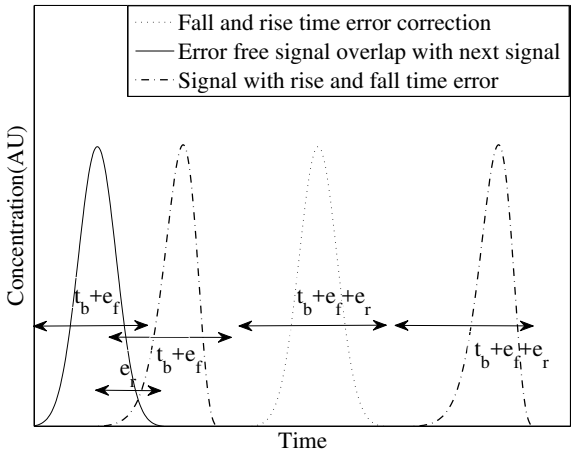


Figure 3.9: Illustration of Rise time error correction

The time period between the end of the i^{th} signal and the start of the $i + 1^{th}$ signal at the receiver represents the i^{th} message. Any deviation from the estimated fall-time alters the stop of the current message. Such an error in fall-time can be corrected by a proper choice of the sampling point. Assuming all other processes to be without error, it is sufficient to start measuring the time period in the rise phase of the receiver response and stop measuring upon the onset of the next rise phase. On subtracting t_b from the total time, the actual message is retrieved. The fall-time error is thus absorbed in the time measurement phase. Such a correction can lead to inter-symbol interference (ISI). The first 2 output signals in Figure 3.8 illustrate interference between signals due to fall-time error in signal 1. To overcome ISI, the bit period is increased from t_b to $t_b + e_f$. The last 2 output signals in Figure 3.8 have an increased bit period thus overcoming ISI.

Rise-Time Error Correction:

The fall-time error correction was based on the assumption that all other timing components are error-free. An accurate ramp-up phase is thus essential in correcting fall-time error. If the propagation delay is error-free, the time at which the leading edge of signal reaches the receiver is error-free. Assuming that the propagation delay is error-free, the response of the receiver is extrapolated to identify the time at which leading edge of signal reached the receiver. The receiver adds the difference between the actual and estimated times of arrival to its measure. In order to remove ISI, the bit period is further increased from $t_b + e_f$ to $t_b + e_f + e_r$. Figure 3.9 illustrates rise-time error and correction. The rise-time error in signal 1 causes interference between signals 1 and 2. Increase in bit period resolves this as seen in third and fourth signals in Figure 3.9. Thus, both rise and fall-time errors are corrected by simply increasing the bit period.

Propagation Error Correction:

The propagation delay determines the time at which the leading edge of a signal reaches the receiver, which in turn conveys the start of a message. Therefore, error in the propagation time is corrected by introducing redundancy in the message as in the simple error correction

scheme with the minimum distance between messages being $4e_d$ instead of $4(e_d+e_f+e_r)$. If the first signal in a communication is error-free, it is possible to decode with zero error for a reduced minimum distance of $2e_d$ as every signal is corrected based on the received and decoded messages i.e., if the $start$ signal is received correct, only the $stop$ signal can be erroneous. Since we decode with 100% accuracy, the error introduced is predicted and the $stop$ is adjusted such that the error does not propagate. The transmission of first signal is restricted to slots of width one bit period ensuring an error-free start signal. In the following sections we assume the first signal to be error-free. The data-rate incorporating smart error correction mechanisms is as follows:

$$R_{tec} = \frac{n}{t_b + t_{in}}. \tag{3.2}$$

$TEC\text{-}SIMPLE$ performs error correction by multiplying each message by $2(e_d+e_f+e_r)$ enabling upto $e_d + e_f + e_r$ error correction. Therefore, the information delay(t_{in}) is

$$t_{in} = \frac{i(2(e_d + e_f + e_r)f_c + 1)}{f_c} : i \in \{0, 1...2^n - 1\} \tag{3.3}$$

$$R_{se} = \frac{n}{t_b + t_{in}}. \tag{3.4}$$

where, R_{se} is the data-rate achieved using $TEC\text{-}SIMPLE$. Employing $TEC\text{-}SMART$, each message is multiplied by $2e_d$ while one bit period is increased from t_b to $t_b + e_f + e_r$. The information delay in this case is given as,

$$t_{in} = \frac{i(2e_d f_c + 1)}{f_c} : i \in \{0, 1...2^n - 1\}, \tag{3.5}$$

$$R_{st} = \frac{n}{t_b + e_f + e_r + t_{in}}. \tag{3.6}$$

where, R_{st} is the data-rate achieved using $TEC\text{-}SMART$ with only error differentiation.

Differential Coding (DC)

From Equation (3.3), it is evident that while curtailing the impact of error has a distinct

benefit on the performance of TEC, the impact of t_b still remains as-is. We thus propose a differential coding (DC) mechanism that leverages correlation between the values of consecutive messages to amortize the impact of t_b. The messages at the source are assumed to be independent and identically distributed. Dependence is introduced by taking the differences of pairs of adjacent messages such that every message in the new sequence is smaller in value compared to that of the original. Since the message is encoded in time, the transmitted values cannot be negative. A sequence of *m* messages is hence arranged in increasing order, and a new sequence constituting differences between adjacent values is formed so that each element in the new sequence is positive and smaller than its value in the original sequence. Since the ordering of elements in the original sequence is altered by virtue of the rearrangement, the actual order must be transmitted as a separate message. If a table of different orders is shared by the end systems, where the table has all possible orders for m messages (i.e., $m!$ entries), a message of size $\lceil \log_2 m! \rceil$ bits is required to transmit the order. Consider an example to understand the aspects of DC. Let the messages to be transmitted by the source be 10(0), 30(1), 5(2), 25(3), 3(4) where the numbers in the bracket denote the position of the message in the sequence. Differential coding is performed in 2 steps. In step 1, the messages are arranged in increasing order. Here, in this example it is 3(4), 5(2), 10(0), 25(3), 30(1). The ordered messages are then passed through differential encoder block that takes difference of adjacent messages giving an output 3, 2, 5, 15, 5 for the above example. Since the messages are arranged in increasing order, the sequence at the output of differential encoder contains only positive values. The position of corresponding order in the table maintained by end systems is transmitted as another message. Let us say the order 4,2,0,3,1 is at position 10 in the table. In this example, the total delay is "40" clock ticks+6t_b as against the "73" clock ticks+5t_b without coding. The number of clock ticks per message is reduced with the use of DC that in turn translates to a higher data-rate.

The sum of elements in the new sequence is equal to the largest element in the original sequence and hence the total waiting time is the sum of the waiting time to transmit the

largest message in the sequence and the corresponding ordering. Let $M = \{m_1, m_2...m_m\}$ be the sequence of messages to be transmitted. The information delay per sequence M, is

$$t_{dc} = \frac{(max(M) + j)(2e_d f_c + 1)}{f_c} : m_i \in \{0, 1...2^n - 1\} \ j \in \{0, 1...m! - 1\}$$
$$R_{dc} = \frac{mn}{(m+1)(t_b + e_f + e_r) + t_{dc}}. \tag{3.7}$$

where, $m_i \in M$ and R_{dc} is the data-rate achieved using DC.

The receiver has to wait till the end of sequence to receive all m messages. Thus, the delay in DC is higher than that in $TEC\text{-}SMART$ without coding but is close to that of OOK. For an n-bit message, OOK takes nt_b time units while DC transmits mn bits in a maximum of mt_b+ t_{in} time units. The delay in DC is close to nt_b units if m is close to n (as $t_{in} \ll t_b$). It has been observed that m is close to n over different values of t_b.

Piggybacked Ordering ($\mathbf{DC_P}$):

Recall that DC adds one extra message per sequence to convey the ordering of messages in the sequence. DC_P is an optimization technique that eliminates the extra message in DC for conveying the ordering of messages. We refer to this variant as $TEC\text{-}SMART(DC_P)$. To keep the number of signals equal to the number of messages, the order is conveyed embedded within the message. Thus, one pair of (bit period + delay corresponding to order) is eliminated at the cost of increased waiting time per message. Every message (the difference) is multiplied by a constant k_1 and a portion of the ordering information is added. Redundancy in information delay and bit period is then introduced to the resultant message for error correction. The receiver, after performing error correction divides the number by the same constant k_1 so that the quotient is the message and the remainder is the portion of ordering. In this fashion, the receiver is able to recreate the ordering message that is embedded in the data messages. The order embedded in each message is k_2. The

information delay in case of DC_P is,

$$t_{dcp} = \frac{(max(M)k_1 + k_2)(2e_d f_c + 1)}{f_c} : m_i \in \{0, 1....2^n - 1\}$$
$$R_{DC_P} = \frac{mn}{m * (t_b + e_f + e_r) + t_{dcp}}. \tag{3.8}$$

where, $m_i \in M$ and R_{DC_P} is the data-rate achieved using DC_P.

The constant k_1 is chosen such that $log_2 k_1 \geq \frac{\log_2 m!}{m}$ i.e., the constant should be able to indicate the number of extra bits per message to represent the order. Considering m = 8, the number of bits required to represent $8!$ is 16 and hence 2 *bits per message* making $k_1 = 4$. The larger the value of k_1, the higher the waiting delay per message. An optimization to choose the best value of k_1, given t_b and m must be performed.

DC for unbounded noise - $\mathbf{DC_U}$:

In this section, we analyze $TEC\text{-}SMART$ in the case of unbounded noise. We propose an optimization to detect error in an unbounded noise channel. When noise distribution is unbounded, it is not possible to achieve 100% error correction. We propose DC_U as an optimization that can detect error in case of unbounded noise. DC_U gives a percentage of correctable, detectable and undetectable error for a given noise distribution. In the rest of the paper, we refer to this variant as $TEC\text{-}SMART(DC_U)$.

As described in Section 3.4, $TEC\text{-}SMART(DC_P)$ requires the sender and receiver to share a list of ordering. For a sequence of m messages, a list of $m!$ entries is shared by sender and receiver. The location of order in the list is appended to the actual message. Noise in the channel alters the location of the order and not the actual ordering. As every received location maps to a valid order, a timing error more than ϵ cannot be detected.

$TEC\text{-}SMART(DC_U)$ detects errors by appending the absolute ordering to the message. In order to represent the order of m messages, each message requires an additional $\log_2 m$ bits. The order in each message is distinct and takes only values from 1 to m. Each message in the new sequence is then multiplied by 2ϵ. If the error is greater than ϵ, the order

appended is changed. Absence of m unique order at the receiver indicates an uncorrected error. DC_U also avoids the need for a list of order to be shared by sender and receiver. No extra memory is required. Thus, if an error greater than ϵ is added to the message, the order as decoded by receiver will not have m distinct numbers thus indicating the presence of an error. In the following conditions, error detection is not possible:(a)Error in each message such that there are m distinct orders but at different positions (b) Large enough ϵ such that order still remains but message is altered.

For a sequence of m messages, there are $m!$ distinct ordering, of which only one is correct. There will be $m! - 1$ possibilities of wrong reception with DC_U. But the total number of erroneous reception can be m^m . Of the m^m possibilities, $m! - 1$ cannot be detected. The rest can be detected. $\frac{m!-1*100}{m^m}$ gives the percentage undetectable error. The choice of ϵ determines the percentage of correctable error and choice of m determines the percentage of detectable error.

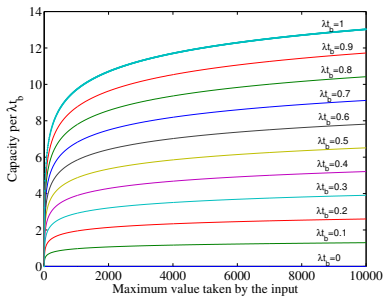
Thus far in this section we have presented $TEC\text{-}SMART$, a communication approach to improve data-rate performance of bacterial communication under non-zero error conditions. In the following sections, we use both theoretical and numerical analysis to evaluate $TEC\text{-}SIMPLE$ and $TEC\text{-}SMART$.

Capacity Analysis

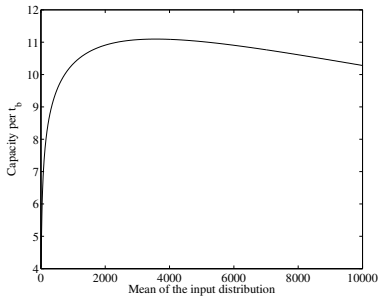
Capacity of a channel is given by the maximum mutual information $I(X : Y)$ between input X and output Y, maximized over all input distributions.

$$C = \lambda \max_{f_X(x)} I(X : Y) \tag{3.9}$$

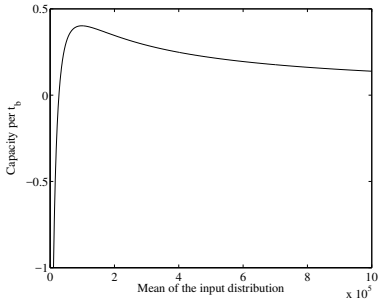
where, $\lambda = \frac{1}{E(Y)}$ is the inter-arrival rate at the receiver. To the best of our knowledge, existing works do not characterize the channel delay of a molecular communication system. We broadly classify channel delay into *bounded* and *unbounded* noise. Among bounded



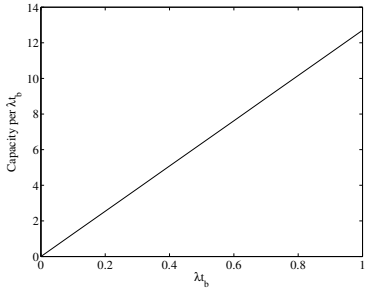
(a) Bounded channel delay with peak-constrained input



(b) Bounded channel delay with mean-constrained input



(c) Unbounded channel delay with mean-constrained input



(d) Bounded channel delay for a given input mean

Figure 3.10: Theoretical analysis of capacity

noise distributions, uniform distribution results in lowest data-rate as all delay components have equal probability. Following queueing theory, exponential service distributed timing channel provides the worst case data-rate performance. We derive the maximum achievable data-rate for uniform and exponential distributions of channel delay.

Uniform Distribution:

Let N be the channel delay. N is uniformly distributed with mean t_b. $N \sim U(t_b - \epsilon, t_b + \epsilon)$. Since information is conveyed in time intervals, there is no parameter analogous to signal power [50]. Therefore, constraint on the input can be mean or peak. Let X and Y be the inter-arrival delays at the sender and receiver ends respectively. Let $x_1 \in X$ be the message to be transmitted. Due to the response time at receiver, the receiver observes $y_1 \in Y$ as $y_1 = x_1 + N_1$, where $N_1 = t_b + n_1$ is the error introduced by the channel and t_b is the average time required by bacteria to respond to a signal. Upon reception, the receiver

subtracts t_b from the observed time and the received message is $y_1 - t_b$. Thus, the system can be modeled using the following equation,

$$Y = X + N - t_b \tag{3.10}$$

We derive the capacity of timing channel using differential entropy of Y and N.

$$I(X:Y) = h(Y) - h(Y|X) \tag{3.11}$$

$$= h(Y) - h(X+N|X) \tag{3.12}$$

$$h(X+N|X) = h(N|X) \text{ , as X and N are independent} \tag{3.13}$$

$$I(X:Y) = h(Y) - h(N|X) \tag{3.14}$$

$$= h(Y) - h(N) \tag{3.15}$$

Case 1: Peak-Constraint:

X $\in$ [0,t_x]. Since X and N are bounded, Y is also bounded. [74] shows that among all bounded distributions, uniform distribution is the entropy maximizing distribution. $Y \sim U(-\epsilon, t_x + \epsilon)$. The differential entropy of uniform distribution is given by, $h(Y) = \ln(t_x + 2\epsilon)$ and $h(N) = \ln(2\epsilon)$. Substituting in Equation 3.11,

$$I(X:Y) = \ln(t_x + 2\epsilon) - \ln(2\epsilon) \tag{3.16}$$

$$C \leq \lambda \ln \frac{t_x + 2\epsilon}{2\epsilon} \tag{3.17}$$

where C is the capacity per average delay of channel and λ=$\frac{1}{t_b + E(Y)}$ is the average inter-arrival rate at the receiver. Since $E(Y) \geq 0$, λt_b varies from 0 to 1. As shown in Figure 3.10a, maximum capacity is achieved when $\lambda t_b = 1$. Also, capacity increases with increasing t_x. Note that λt_b is strictly less than 1, as $E(Y) = E(X)$ and $E(X) \geq 0$. The different colors in 3.10a denote different values of λt_b. For a given t_x, depending on the error correc-

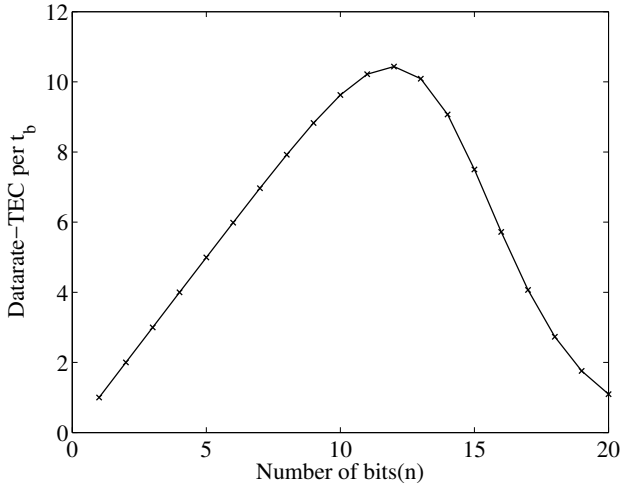


Figure 3.11: Simulation based capacity

tion mechanism and modulation, the system approaches a certain ratio of λt_b. The higher the value of λt_b is, the better the algorithm is, in achieving the maximum data-rate. The smaller the value of $E(X)$, higher the ratio λt_b i.e., for small values of $E(X)$, $E(Y) \approx t_b$. The delay at the receiver is thus dominated by t_b leading to an increased data-rate.

If $\epsilon << t_x$, then an approximation for entropy maximizing input distribution can be derived. Assume $X \sim U(0, t_x)$. We assumed $N \sim U(t_b - \epsilon, t_b + \epsilon)$. The distribution of sum of 2 independent random variables is the convolution of 2 distributions. Here, both X and N are uniformly distributed. Convolution of these 2 uniform pulses gives a trapezium. The slope of the sides of the trapezium is very high if $\epsilon << t_x$, which we can approximate to a uniform distribution. Hence, for peak constrained input in a uniform noise distribution channel such that $\epsilon << t_x$, uniformly distributed input maximizes channel capacity.

Case 2: Mean-Constraint:

$E(X) \leq k$ where k is an arbitrary constant. The mean of the input distribution is constrained. Since $Y = X + N - t_b$, $E(Y) = E(X)$, Y is also mean-constrained. Note that $Y + t_b$ is a measure of time and is positive. Among all mean-constrained, positive distributions, exponential distribution gives the maximum entropy. Thus, capacity is upper bounded when $Y + t_b$ and hence Y follows exponential distribution. Since entropy does

not change with linear translation, $h(Y) = 1 - \ln \frac{1}{E(Y)}$. Similar to case 1,

$$I(X:Y) = 1 - \ln \frac{1}{E(Y)} - \ln(2\epsilon) \text{ as E(Y)=E(X)}, \tag{3.18}$$

$$C \leq \lambda(1 + \ln \frac{E(X)}{2\epsilon}) \tag{3.19}$$

Total delay per reception is $E(Y) + t_b$. Thus, the capacity per average delay of channel is,

$$C \leq \lambda t_b(1 + \ln \frac{E(X)}{2\epsilon}) \tag{3.20}$$

Figure 3.10b shows the capacity as a function of mean of the input with $t_b = 435$min and $\epsilon = 0.6$s. With increasing mean, the capacity increases to a maximum and then decreases. Till the peak, total delay is dominated by t_b after which, the delay increases linearly whereas the number of bits represented increases logarithmically. Thus the net data-rate decreases.

Exponential Distribution:

In case of unbounded distribution, peak-constraint for input is not tractable. Therefore, we consider a mean-constrained input. Let N $\sim$ Exp($\frac{1}{t_b}$). Following the case 2 of uniformly distributed noise, Y should follow exponential distribution with mean $E(Y) = E(X)$.

$$I(X:Y) = 1 - \ln \frac{1}{E(Y)} - 1 + \ln \frac{1}{t_b} \tag{3.21}$$

$$C \leq \lambda \ln \frac{E(X)}{t_b} \tag{3.22}$$

where C is the capacity per average delay of channel. Following the theoretical analysis, the maximum achievable data-rate under different constraints on input distribution for uniform and exponential noise distribution has been derived. The performance of proposed error correction scheme along with timing modulation is compared against channel capacity. The simulation results do not include the differential encoder block as data-rate across channel is compared. Figure 3.11 shows the data-rate performance based on simulation results.

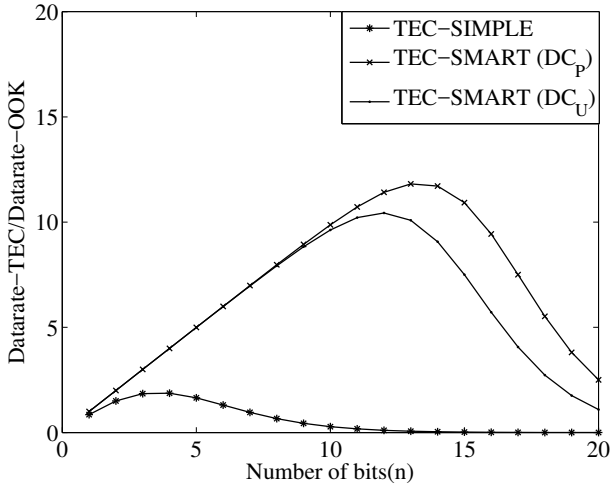


Figure 3.12: Performance of $TEC\text{-}SMART$ and $TEC\text{-}SIMPLE$ with varying frame size n

The results show that the proposed error correction has $10.5X$ improvement over OOK with peak constraint on input at 2^{12}. The input and the noise were uniformly distributed. Under the given conditions, achievable capacity is $11.7X$ over OOK. The data-rate of the proposed solution is 90% of that of the maximum achievable data-rate.

Numerical Analysis

Evaluation

We now perform numerical analysis of Equations (3.1) to (3.8) using MATLAB. The specific values for the parameters and the ranges for parameters used are driven by the experimental results presented in Section 3.1. Unless otherwise specified we use the following values: t_b =435 min, t_d =6 sec, $e_f + e_r = 0.1t_b$, $e_d = 0.1t_d$. Since the performance of $TEC\text{-}SMART$ is dependent on the message size, the bit period, error introduced by the channel and the clock rate, we study the sensitivity of its performance to these different parameters. We present only relative performance results for TEC and its variants with respect to OOK. Every data point is obtained by taking an average of data-rate corresponding to all messages of frame size n.

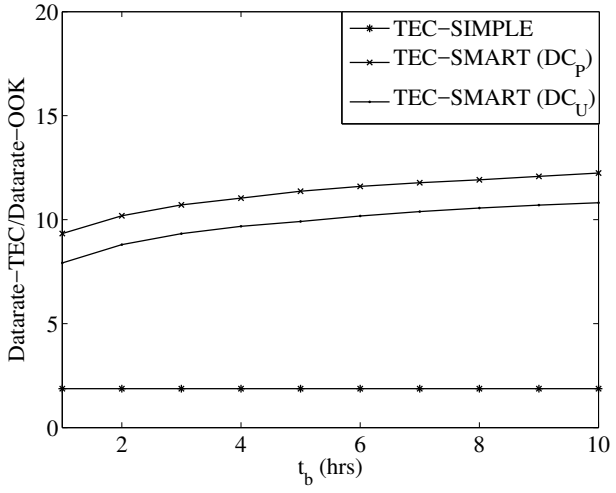


Figure 3.13: Performance of TEC-$SMART$ and TEC-$SIMPLE$ with varying bit period t_b

Frame Size: Unlike other modulation techniques, the data-rate of TEC varies with the frame size n. The total delay for a transmission varies with the absolute value of the message. For small values of *n*, information delay $t_{in} << t_b$. Thus, the data-rate increases with increasing *n*. Once t_{in} is comparable to t_b, the data-rate begins to decrease as t_{in} starts dominating. The relative data-rate performance of TEC is presented in Figure 3.12. The performance of TEC-$SMART(DC_U)$ is for an unbounded channel delay distribution. The goal of TEC-$SMART(DC_U)$ is to detect error in the presence of unbounded noise and hence the maximum data-rate achievable is smaller than TEC-$SMART(DC_P)$, which cannot correct or detect any error greater than e_d.

Bit Period: Figure 3.13 presents the data-rate performance for TEC, TEC-$SMART(DC_P)$ and TEC-$SMART(DC_U)$ for different bit periods. The value of t_b is varied from 1 to 20 hours. It can be observed that, while TEC is impacted heavily by an increase in t_b, TEC-$SMART(DC_P)$ and TEC-$SMART(DC_U)$ is considerably more resilient to larger values of t_b. This is due to the amortization of the t_b overhead over multiple messages.

Frequency: Figure 3.7d shows an increase in the data-rate with increasing clock frequency. With the introduction of error in the system, the clock rate loses its significance. Recall that the transmitter and the receiver measure the number of e_d time units between signals. How-

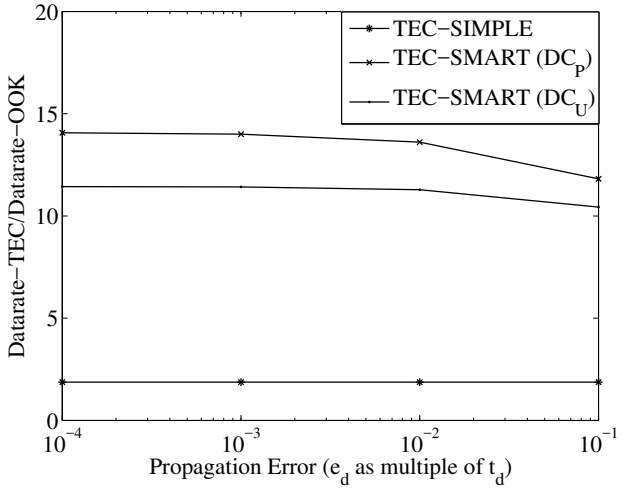


Figure 3.14: Performance of $TEC\text{-}SMART$ and $TEC\text{-}SIMPLE$ for varying Propagation Error

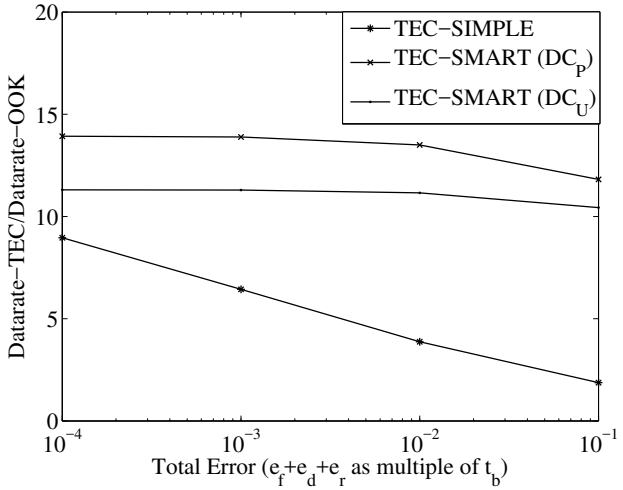


Figure 3.15: Performance of $TEC\text{-}SMART$ and $TEC\text{-}SIMPLE$ for varying Total Error

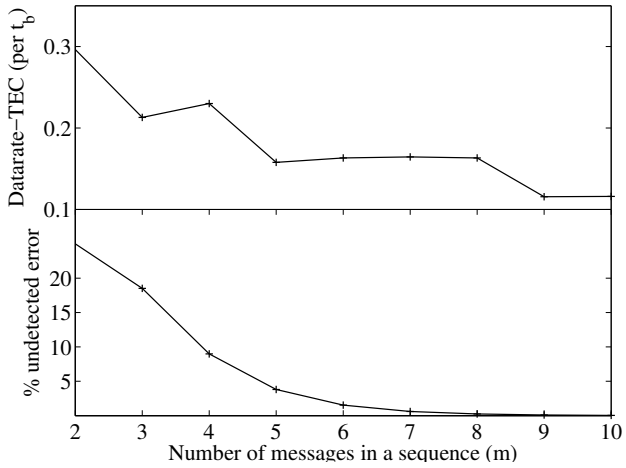


Figure 3.16: Effect of Number of messages per sequence on datarate : Exponential Channel Noise

ever high the clock rate is, the time slot is now in terms of error and hence the data-rate performance does not change with frequency once the error correction is introduced.

Error : We proposed $TEC\text{-}SMART$ as a better error correction strategy. $TEC\text{-}SMART$ considers both bounded and unbounded error and proposes strategies to detect uncorrected error with high probability in case of unbounded error. We analyze the performance of $TEC\text{-}SMART$ under bounded and unbounded error for varying error conditions.

Bounded Error: $TEC\text{-}SMART$ is explicitly designed to handle error conditions better by virtue of its error curtailing and differentiation mechanisms. Thus, the increase in *rise-time error* and *fall-time error* has minimal impact on the overall performance of $TEC\text{-}SMART$. In this section, we analyze the results in a bounded error. As seen in Figure 3.14, $TEC\text{-}SMART(DC_P)$ can deliver a data-rate of over 10x even when the total error is large ($0.1t_b$+e_d). Data-rate with respect to varying error components is presented in Figures 3.14-3.15. Overall, the results demonstrate the better error resiliency exhibited by $TEC\text{-}SMART(DC_P)$.The data-rate delivered by $TEC\text{-}SMART(DC_U) < TEC\text{-}SMART(DC_P)$ but the former can detect error greater than e_d.

Unbounded Error: In the case of positive valued unbounded channel delay, exponential distribution can be considered as a general case, similar to Gaussian distribution in energy

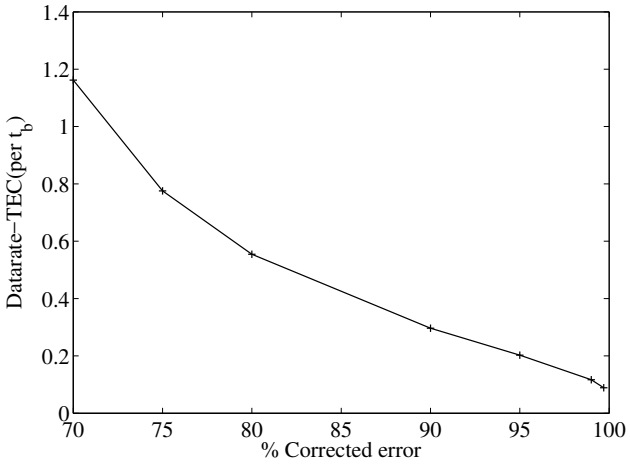


Figure 3.17: Effect of Propagation error on datarate : Exponential Channel Noise

based communication. Figure 3.16,3.17 shows the performance of $TEC\text{-}SMART(DC_U)$ under exponential channel delay. The percentage of correctable error is increased by increasing ϵ but this reduces the data-rate due to the increase in redundancy. Figure 3.17 shows the decrease in data-rate with increasing ϵ. The following analysis is used to estimate ϵ for a given % of error correction. Let a be the fraction of error to be corrected and $f(x)$ be the probability distribution of exponential error. For e.g, $a = 0.9$ for 90% error correction.

$\int_0^{\epsilon} \mathrm{f(x)}\,\mathrm{d}x = a, F(\epsilon) = a$ where, F(x) is the cumulative distribution function

For an exponential distribution,$1 - e^{-\lambda\epsilon} = a, \epsilon = -\lambda(ln(1-a))$

Figure 3.16 shows the variation of data-rate and percentage of undetected error with increasing m. The correctable error is set to 90%. The value of ϵ to be multiplied to the message is obtained from the probability distribution of channel noise as explained above.

Discussions and Future Work

- *Dynamic System :* In the analysis of the network topology and the design and evaluation of TEC, we assumed the nodes to be static. The time elapsed between signals does not change over a period of time. However, if the transmitters and/or the receiver is mobile, the distances between the transmitter and the receiver thus affecting the propagation and the time elapsed between the signals. Implementation of TEC and TEC-SMART in a mobile system is an open challenge.

- *Propagation Error :* The first component of TEC-SMART is Error-Differentiation which decomposes ϵ into three components of error. It corrects for propagation error by increasing the minimum distance between messages by $2e_d$, where e_d is the propagation error. For short distance, propagation error is of the order of few seconds and does not affect the overall delay of the system. As the distances increases, propagation time increases and therefore e_d increases, in turn decreasing the throughput performance of TC-SMART. A more detailed analysis of the practical distances between transceivers and modeling of the propagation error is a part of our future work.

- *Longevity of Engineered Bacterial Receiver and Transmitter :* In the molecular communication network, the transmitter, the receiver are engineered bacterial colonies. In nature, bacterial colonies evolve over generations and acquire new capabilities over time. The fidelity of the engineered bacterial colonies is still an open challenge to implement a molecular communication system. The maximum duration for which the bacterial colonies perform the engineered functionalities must be studied. The longevity of bacterial colonies must be studied both in isolation and in the presence of other natural and engineered bacterial colonies.

- *Capacity Analysis :* In the capacity analysis of the timing channel presented above,

we derived the capacity of the timing channel for uniform and exponential distribution of channel noise. Uniform and exponential distributions were chosen as the distribution for bounded and unbounded error respectively, as these distributions have the worst-case impact on the network performance. A more realistic modeling of the channel noise model will benefit the capacity analysis of a realistic molecular communication system. In the design and analysis of TEC and the capacity of timing channel, we assumed that the demodulation error from the response of receiver colony is bounded and uniform. An in-depth analysis and modeling of the error in the receiver response is a part of the future work in the capacity analysis of timing channel. A fundamental limit on the capacity of timing channel will be useful in the future of variants of TEC and other timing based communication algorithms.

- *Clock variability :* The promise of TEC is achieved by offloading signal processing to clocks at the transceivers. The variability in the clock rates at the transmitter and receiver was assumed to be negligible in the design and evaluation of TEC. Study and design of robust TEC that can correct bit errors in the presence of counting errors is a part of the future work.

CHAPTER 4

ADMA: AMPLITUDE-DIVISION MULTIPLE ACCESS FOR BACTERIAL COMMUNICATION NETWORKS

In the pathogen detection system considered, by introducing more than one sensor, real-time monitoring of a variety of pathogens is possible. In the design of TEC, we considered a single biolink with one transmitter and receiver. Majority of prior research in MC have focused on channel and system modeling [42, 43], capacity derivation [44, 45, 46], modulation techniques [47, 48, 49] and analysis of channel and inter symbol interference [46, 50, 51]. These studies focus on a single link and do not consider the challenges in implementing the algorithms in a real-life environment. In this work, we consider a star topology with multiple sources and a single receiver, as shown in Figure 4.1. This topology is most commonly seen in sensor networks, where multiple sensors communicate with a single receiver/sink. The sensors broadcast information and do not require a destination address. The receiver, on the other hand, receives a cumulative signal from multiple sources, making it necessary to have an efficient addressing mechanism that uniquely identifies the sources.

Modulation techniques specifically targeting MC have been developed by a number of researchers [47, 46, 48]. Concentration shift keying [47] and molecule shift keying [46, 48] are two well-known methods that encode information in the concentration (amplitude) of the signal and the molecule type respectively. With the growth of network sizes, there is a need for algorithms that perform addressing, medium access control (MAC), routing and reliability for networks and not just links. In [75], a distance-based addressing that estimates the distance between transmitter and receiver using beacons, propagation delay, and path loss is proposed. [75] establishes a coordinate system from the distances measured and the molecules move to the desired location. This addressing mechanism assumes that the transmitter can identify its location in the channel and guide molecules to a particular

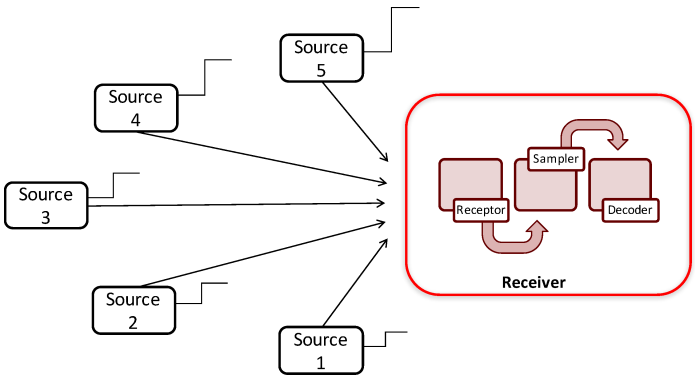


Figure 4.1: Network Setup

direction. Developing such a transmitter using biological circuits is highly challenging. Extending this scheme to more than one link will be challenging as it requires accurate channel estimation. [51] proposes a MAC using *molecule type*, wherein each transmitter communicates with a distinct molecule, and thus do not interfere with other transmitters, analogous to frequency division multiplexing. We explore this approach in detail in the following section.

In this work, we focus on *source addressing*, an addressing mechanism that can distinguish multiple sources communicating with a single receiver in a star topology as shown in Figure 4.1. We consider the experimental system used to validate OOK and TEC; signaling chemical molecules propagate through a microfluidic channel and trigger the receiver to generate Green Fluorescent Protein (GFP), which is then sampled, demodulated and decoded. Each source uses On-Off-Keying (OOK), a simple modulation technique that transmits a rectangular pulse $m(t)$, as shown in Equation (5.1), with concentration A for T seconds (bit period) to transmit bit 1 and an absence of signal for T seconds to transmit bit 0.

$$m(t) = \begin{cases} A & 0 \leq t \leq T \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Types of Addressing

Address Fields:

In traditional communication systems, it is common to allocate a fixed number of bits in the packet header for addressing, e.g. the IP address and MAC address fields. In bacterial communication networks, on the other hand, very high processing delays at the transceiver nodes result in extremely low data rates (order of 10^{-5} bits per second [26]). Overheads in the form of address fields can result in additional per-frame delays as well as a decreased per-user throughput.

With the use of modulation techniques such as concentration shift keying [47], the number of symbols (signals) required to transmit an address can be reduced. However, a significant reduction in the network throughput cannot be avoided without fully eliminating the use of address fields. We propose *Embedded Addressing*, an addressing mechanism that eliminates the need for address fields by embedding the source address in the transmitted signals. It uses the unique characteristics of molecular signals to identify the senders (sources).

This is a local addressing mechanism, i.e., the address embedded is local to the network and not the sender's unique global address. We assume that each source has already been assigned a unique global address[1]. The receiver uses only the received signal and its knowledge of the characteristics of the source signals to identify them locally. It is worth mentioning here that code division multiple access (CDMA) is also a technique that uses embedded addressing by assigning a unique pseudo-random spreading code to each source. However, it does not provide a solution for slow networks (that are common in MC) because of its use of spreading codes which further decreases the data rate of the network.

In MC, we identify the following key characteristics of a molecular signal that allow us to embed the source addresses in the transmitted signals without compromising the network throughput: *molecule type*, *signal duration*, and *signal amplitude*. We next elaborate on

[1] Assignment of global addresses is out of the scope of this work.

Table 4.1: Source Addressing Mechanisms

Addressing Mechanism	Strength	Weakness
Address Fields	Global Address	Increased Delays, Reduced Throughput
Molecule Type	Increased Throughput, Fair	Not Scalable, Complex Receiver Circuits
Pulse Duration	Simple Transceiver Circuits	Reduced Throughput, Introduces Unfairness
Amplitude	Simple Transceiver Circuits, Scalable, Fair	Network Size Limited by Receiver

each of these properties.

Molecule Type: The address is embedded in the signaling *molecule type* with the same amplitude and duration for all sources. Each source is assigned a unique molecule, and the receiver must be capable of receiving all the molecules, allowing all the sources equal access to the receiver without contention. Thus, source addressing with molecule type also solves the MAC problem.

The other two characteristics, pulse amplitude and duration, allow all the sources to transmit the same molecule. The receiver accepts only one type of molecule, simplifying the receiver and transmitter designs.

Pulse Duration: The amplitude of the signal and the molecule type are fixed across sources, with the address embedded in the signal duration. Each source is assigned a unique pulse duration. When a source has bit 1 to transmit, it transmits a signal with a given amplitude and the duration assigned to it as the address. The sources are assigned distinct durations, which leads to increased latency. The per-frame delay of each source is different from the others, leading to unfair throughput and increased network delays.

Signal Amplitude: In electromagnetic communication, amplitudes have been used for modulating the signal. Here, we consider using amplitudes as the address. Each source is assigned a unique amplitude, and transmits its signal with the assigned amplitude for a duration fixed for the network. The receiver maps the received amplitude to the respective source. When multiple sources transmit at the same time, the receiver receives the sum of

amplitudes. To identify each source, the receiver must determine the individual amplitudes for all received sums.

Table 4.1 summarizes the strengths and weaknesses of the above mechanisms, and we conclude that embedding the address in the amplitude of the signal is the most efficient among the four source addressing mechanisms.

Experimental Validation of Amplitude Differentiation

Due to the challenges in the design of state-of-the-art microfluidic system [76], the proof-of-concept presented in this work extrapolates results from a system with one source and one receiver. We extrapolate experimental validation in the following steps. Using the experimental setup, we verify that a receiver can distinguish between amplitudes from the GFP response of the receiver bacteria. [16] proved experimentally that the response of the receiver bacteria is distinct for distinct input amplitudes. We use the model developed in [16] that was validated with experimental results to model the response of receiver bacteria to a molecular signal.

In this work, we implement the numerical inverse of the above model as a demodulator. In the *inverse model*, the receiver response is the input and an estimate of the molecular signal that triggered this response is the output. We use a model that was experimentally validated to generate the response of receiver and an inverse module that demodulates the signal from the response of the receiver. We assume N parallel point-to-point channels from N sources to 1 receiver as shown in Figure 4.1. Therefore, the molecular signal reaching the receiver is the sum of amplitudes of the signals transmitted by each source. The cumulative signal from the individual sources becomes the input to the receiver model. The response of the receiver to the cumulative signal simulates a multiple-source-single-receiver topology. The modulation and channel error of each channel is handled individually. In this work, we consider a system in which genetically-engineered bacterial populations are receivers connected through microfluidic pathways. Microfluidic pathways allow for dynamic changes

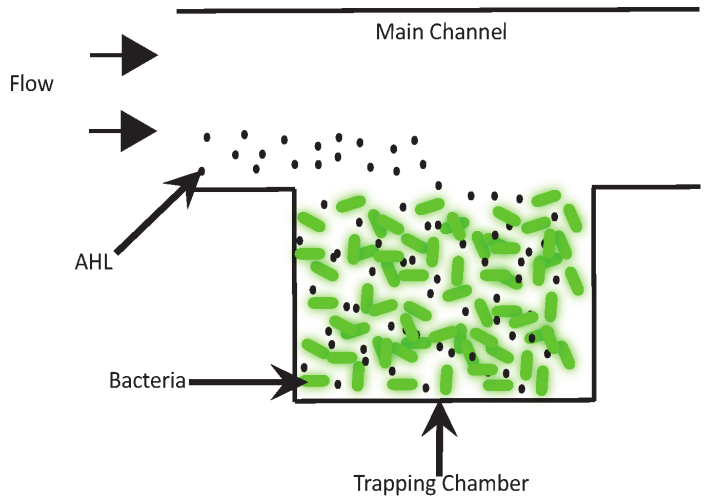


Figure 4.2: Microfluidic channels in direct fluidic contact with trapping chambers housing bacteria.

in media composition. The constant stream of media keeps the bacteria in ideal growth conditions, eliminating growth phase dependent variables from the experiments. The bacteria density does not change during the experiment. The bacteria are first seeded into the trapping chambers and grow until the chamber is filled. The chamber is in direct fluidic contact with the main channel, which has constant flow providing nutrients to the bacteria and removing excess bacteria. This means the density of the chamber contents remains constant.

The bacteria used in all experiments were a genetically-engineered strain of DH5α *E. coli*. Methods and functionality of the bacteria and the microfluidic device fabrication and specifications can be seen in previous works [26, 16], but will be briefly mentioned here. In this system, the chemical stimulus is autoinducer N-acyl homoserine lactone (AHL), and the bacterial response is the expression of GFP. To fabricate the microfluidic devices, we utilize standard soft lithography [77] resulting in a polydimethylsiloxane (PDMS) device bonded to a glass coverslip. The design consists of the main channel which has direct fluidic contact with adjacent chambers that house the bacteria for the duration of the experiment, as seen in Figure 4.2.

Figure 4.2 shows the receiver bacteria. The "Main Channel" is the channel through

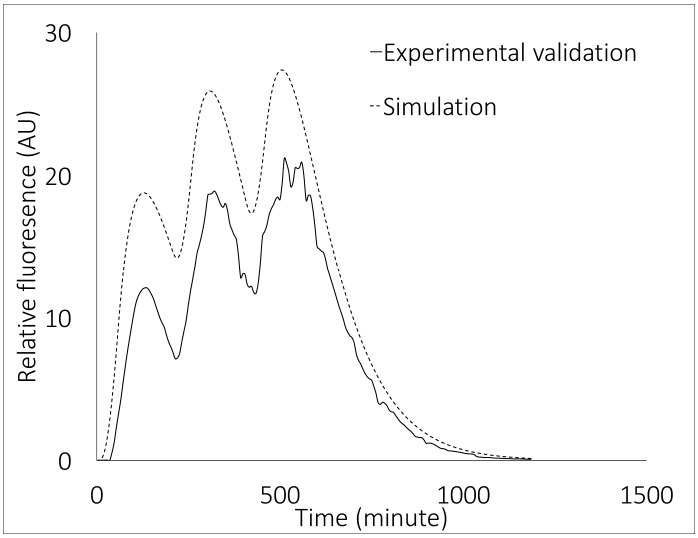


Figure 4.3: Bacterial Receiver Response

which molecules carry information. After initial loading of the chip, the bacteria were allowed to populate the chambers for 24 hours to reach capacity. During this time the bacteria were supplied with a constant flow of 2xYT lysogeny broth (LB) at 100 $\mu l/hr$. Once the bacteria had filled the chamber, flow rate was increased to 360 $\mu l/hr$. One syringe was used for LB (at 350 $\mu l/hr$), while the second (10 $\mu l/hr$) was used to deliver varying concentrations of AHL. An "AHL pulse" is the duration and concentration over which this AHL was delivered to the bacteria chamber. Pulse durations of 50 minutes that we optimized [26] were used for "on" or "bit 1" state; while 50 minutes of no AHL equated to "off" or "bit 0" state. Unless otherwise mentioned, we use the above pulse concentration and duration throughout this work. Fluorescent images were captured once every 10 minutes and post-processed using MATLAB. The intensity of the pixels within the bacteria chamber was averaged and the background fluorescence was subtracted, yielding relative fluorescence (arbitrary units, or AU). Four dynamically programmed signals of *on* and *off* states were delivered to the receiver bacteria populations while their fluorescent outputs were recorded. The following input bit patterns were tested: 1010101010, 1000100010, 1010000010, and 0000000000. The bit patterns are chosen to represent different probabilities of bit 1. Utilizing the receiver response model developed in [16], we compared the experimental results

to the predicted response and found that the model captures the dynamics well. Figure 4.3 shows the response of the receiver bacteria from experiments and simulations to an input of 1000100010 in one trial.

To further test the model's capabilities to demodulate the received signal, we used the experimental GFP results to decode the original AHL input signal by using the inverse of the model proposed in [16]. The decoding efficiencies of the model are, respectively: 90%, 100%, 80%, 100%. Each bit pattern was repeated four times, and the average decoding efficiency we present here is the average over the four repetitions. These experiments are time intensive and challenging due to the nature of bacteria. There are numerous factors that affect the ability of bacteria to process the AHL input and produce the GFP output. We have made great strides in controlling several of these factors, such as temperature, population size, and nutrients, by utilizing a microfluidic device, but several remain out of our control. These uncontrollable factors can fluctuate and cause small variations in the bacterial response, which can lead to lower decoding efficiencies. This is most likely the cause for the 80% decoding efficiency.

Amplitude-Division Multiple Access

Problem Definition

As summarized in Table 4.1, relying on *Address Fields*, *Pulse Duration*, and *Molecule Type* as addressing mechanisms can result in increased network delays, decreased throughput, complex receiver circuits, and throughput unfairness. On the other hand, embedding address in the amplitude of the signal is both fair and throughput friendly, but the maximum number of users is limited by the number of amplitude levels the receiver can distinguish. In the arguments above, we show that embedding the address in the amplitude is best suited for source addressing in a super-slow network, such as a bacterial communication network. Amplitude source addressing requires simple source and receiver bacterial circuits, which

will be described in Section 4.10. [16] demonstrated that the response of receiver bacteria is distinct for each distinct amplitude, making the use of amplitude address practical. However, the receiver response model developed by [16] does not consider the distance between the source and the receiver as the microfluidic system in [16] uses flow channels, i.e., the nutrients and signals are carried to the receiver at a given flow rate. In this case, the distance between the source and the receiver [26] affects only the propagation delay and has no impact on the performance of ADMA.

To this end, we propose *ADMA*, a source addressing mechanism that embeds the source address in the amplitude of the signal transmitted. In this section, we present the goals and challenges of using *ADMA* in a practical system. When multiple sources access the channel at the same time, their signals collide, and as a result, amplitudes are summed up in the channel. In the following section, we present an optimal amplitude assignment that can achieve zero address resolution error. Goals of *ADMA* are,

- to resolve the bits transmitted by sources, given a received amplitude, with minimum error, and
- to maximize the number of users that can be addressed.

Optimal Amplitude Addressing

Amplitude source addressing can be 100% accurate if the receiver can resolve every source given a received amplitude. Addressing and MAC has two main goals: 1) reliable packet delivery 2) throughput fairness. We define the Collision Resolution Error (CoRE) as a metric to measure reliability. CoRE is the ratio of the number of sources identified incorrectly to the total number of sources. An optimal amplitude assignment will achieve zero CoRE. CoRE is determined by the choice of amplitudes and the receiver design. We derive the conditions to achieve zero CoRE and propose an optimal amplitude assignment that achieves zero CoRE. Let b_i be the bit transmitted by i^{th} source with amplitude a_i in a given

time slot.

$$b_i = \begin{cases} 1 & \text{if } i^{th} \text{ source transmits bit 1} \\ 0 & \text{if } i^{th} \text{ source transmits bit 0} \end{cases}$$

When molecular signals collide in the channel, the receiver obtains the sum of amplitudes transmitted $y = \sum_{i=1}^{N} b_i a_i$, where N is the number of sources in the network. If the number of partitions of the received amplitude y is one, i.e., the number of ways in which different a_i can be added to reach a sum y is one, then CoRE will be zero. If the number of partitions of y is greater than one, then CoRE is strictly greater than zero. Table 4.2 shows a sample network with three sources, each transmitting $b_i \in \{0, 1\}$. The amplitudes assigned to sources are $\{1, 2, 3\}$, creating 2^3 possible combinations of bits. We refer to each bit combination as a configuration, as shown in Column 4. The sum of amplitudes corresponding to each configuration is defined as its magnitude, which depends on both the configuration and the amplitudes assigned. Note that the configurations $\{0, 0, 1\}$ and $\{1, 1, 0\}$ have a magnitude of 3, which implies that the number of partitions for *3* in this setup, is two. On receiving an amplitude 3, the receiver must choose from the partitions of 3.

To achieve zero CoRE, the number of partitions of every received amplitude must be less than or equal to one. In other words, the magnitude of each configuration must be unique. This problem is studied in Number Theory as "Distinct subset sum (SSD)". A set is defined an SSD if and only if the sum of every subset of the sequence is unique[78]. An example of an SSD is the binary set, the set of powers of 2, $S = \{1, 2, 4, 8\}$. Each subset has a unique sum as every configuration has a unique magnitude. The majority of research on SSD focuses on finding the limit of the maximum value in a subset sum sequence [79]. In bacterial communication, the range of amplitudes $[R_{min}, R_{\max}]$ that the receiver can distinguish is determined by the receiver circuit design. The receiver circuit thus determines the following parameters of the network.

Table 4.2: Configurations

b_{u1}	b_{u2}	b_{u3}	Configuration(C_i)	Mag(C_i)	$\Pr(C_i)$	Received Amplitude
0	0	0	0,0,0	0	$(1-p_t)^3$	0
0	0	1	0,0,1	3	$(1-p_t)^2 * (p_t)$	3
0	1	0	0,1,0	2	$(1-p_t)^2 * (p_t)$	2
0	1	1	0,1,1	5	$(1-p_t) * (p_t)^2$	3
1	0	0	1,0,0	1	$(1-p_t)^2 * (p_t)$	1
1	0	1	1,0,1	4	$(1-p_t) * (p_t)^2$	3
1	1	0	1,1,0	3	$(1-p_t) * (p_t)^2$	3
1	1	1	1,1,1	6	$(p_t)^3$	3

1. The minimum decodable amplitude at the receiver R_{min}.

2. The maximum receivable amplitude R_{max} beyond which the receiver saturates. If an amplitude greater than R_{max} is transmitted, it is received as R_{max}.

3. The step size of the levels of amplitudes that the receiver can distinguish. R_{max} and step size δ of the amplitudes determines the number of amplitude levels that can be distinguished, i.e., $N = \dfrac{R_{\text{max}} - R_{\text{min}}}{\delta}$ levels. By factoring the step size out and subtracting R_{min}, the amplitudes that can be assigned are integer values $1, 2, 3, \ldots, N$. Therefore, in *ADMA*, integer amplitudes are used to analyze the performance of the network. The addresses proposed here can be used in a network with a step size greater or less than "1" by multiplying the proposed addresses with δ.

Theorem 1. *For a given maximum sum, the set of powers of two (binary set) is an optimum set of amplitudes that render zero CoRE.*

Proof: A set **S** with *n* elements has up to $2^n - 1$ non-empty subsets, hence $2^n - 1$ non-zero sums. To achieve zero CoRE, these $2^n - 1$ sums must be distinct. Each sum is different from another by at least one, so, the sum of all elements is at least $2^n - 1$. A binary set, consisting of powers of 2, satisfies the above condition. A binary set with all powers of 2 is an optimum set of addresses that minimizes CoRE. The maximum number of sources that can be supported by the binary set is related to R_{max}. ■

A binary set can accommodate up to $N_{\log}$ sources, where $N_{\log} = \lfloor \log_2(R_{\text{max}} + 1) \rfloor$. By assigning addresses from the binary set to sources, we also solve the multiple access

problem. When signals from multiple sources with amplitudes from the binary set collide at the receiver, their sum is mapped to a unique configuration, allowing the receiver to decode with zero CoRE. As a corollary of Theorem 1, when $N > \lfloor \log_2(R_{\text{max}} + 1) \rfloor$, CoRE is strictly greater than zero.

Though the binary set achieves CoRE, it limits the number of sources to $N_{\log}$. If $N > N_{\log}$, a set of addresses that can accommodate all the sources and minimizes CoRE is required. We propose a heuristic algorithm to select a set of amplitudes that approaches minimum CoRE, given N and R_{max}. The two major challenges in designing an *ADMA* system that minimizes CoRE are,

- designing a sequence of amplitudes,
- designing a scalable and low complexity decoder.

Components of the *ADMA* Architecture

In this section, we define and describe each component of the receiver architecture of ADMA. The three components of the receiver architecture are sampler, demodulator, and decoder. The response of the receiver bacteria, shown in Figure 4.3, is input to the *sampler*, that samples it to discrete received amplitudes. Sampling and demodulation utilize the inverse of the bacterial receiver response model derived in [16]. The output of the sampler (the inverse model) is the time sequence of the molecular signal samples received. Thus, the inverse model determines the sampler's accuracy. The model developed in [16] is deterministic and does not account for the stochastic nature of receiver bacteria. In order to realize the stochastic nature of receiver response to input chemical signal, we introduce random noise to the "k parameters" of the inverse model. The "k parameters" are the different rate constants of the receiver bacteria (example, GFP expression rate) that define the state of the receiver bacteria. By varying these parameters randomly within a specified range, we implement sampling and demodulation errors in the receiver response.

The samples are then input to the *decoder* to resolve the addresses and bits. We present decoder designs in detail in Section 4.7. The *decoder* outputs a vector of bits, the estimate of bits transmitted by all sources that sum to $y[i]$, the received sample. Thus, the receiver design determines address resolution on receiving a signal, which in turn, determines CoRE. Here, we propose two receiver designs based on the principles of Maximum a posteriori detection: 1. a probabilistic receiver, and 2. a deterministic receiver. The receiver decodes the samples, which are then used to decode the bits. The optimality of the receiver design and the time complexity of the decoder are analyzed at the sample level. The number of samples per bit is pre-determined based on the application and system constraints. Thus, there is no need for coordination or time synchronization between sources, and the sources do not require additional processing to synchronize transmission, i.e., each source can transmit data as and when it has information to transmit. Assuming that all sources always have data to transmit, the receiver is continuously receiving samples. The most recurring value of samples is then used to determine bits. A receiver that maximizes the probability of success of each sample, in turn, maximizes the probability of success of the bit decoded from these samples.

ADMA Receiver Designs

In the previous section, we presented an optimal amplitude addressing that achieves zero CoRE. But, the maximum number of sources N is limited by $\lfloor \log_2(R_{\text{max}} + 1) \rfloor$. As derived in Theorem 1, when N is $> \lfloor \log_2(R_{\text{max}} + 1) \rfloor$, average CoRE will be strictly greater than zero. In a network with $N > \lfloor \log_2(R_{\text{max}} + 1) \rfloor$, at least one amplitude has more than one partition and therefore the receiver design also contributes to CoRE. In this section, we propose two receiver designs and derive an upper bound on the expected number of success in resolving address with each receiver design.

We make the following assumptions about the network in the design and evaluation of the receiver designs. R_{max} is the maximum amplitude that the receiver can uniquely

identify. Each source is assigned a distinct amplitude, and hence up to $R_{\max}$ sources can be accommodated, i.e., $N \leq R_{\max}$. If $N > R_{\max}$, the network is divided into subnets. $S = \{a_1, a_2, \ldots a_N\}$ is the set of amplitudes assigned to sources $u_1, u_2, \ldots u_N$ respectively. The sources always have data to transmit; at a given time, a source is either transmitting bit 1 or transmitting bit 0 and these N sources can be transmitting in one of the 2^N configurations. Consider the example in Table 4.2. Columns 1 to 3 in Table 4.2 are the bits transmitted by sources 1 to 3. Column 5 is the magnitude of the configuration, equal to the sum of amplitudes corresponding to each configuration and Column 7 is the amplitude received when the corresponding configuration is transmitted. Multiple configurations can add up to the same sum. All configurations with the same magnitude are called the *partitions* of that magnitude. Since $R_{\max} = 3$, any amplitude ≥ 3 is received as 3. Here, $\{0,0,1\}, \{0,1,1\}, \{1,0,1\}, \{1,1,0\}, \{1,1,1\}$ are the partitions of 3, as all these configurations are received as 3 by the receiver. p_t and $1 - p_t$ are the probabilities of a source transmitting bit 1 and bit 0 respectively. The probability of a configuration occurring in the channel depends on p_t and the addresses. For example, the probability of C_0 is $(1 - p_t)^N$ as all N sources transmit bit 0, each with probability $1 - p_t$. The probability of each configuration is shown in Column 6 of Table 4.2.

We propose *Probabilistic Receiver* to minimize the error in decoding a configuration and *Deterministic Receiver* to decrease the bit error for individual sources. We also derive an upper bound on the expected number of successful address resolutions for the above two receivers, which is then used to calculate a bound on the throughput performance of ADMA with each receiver.

Probabilistic Receiver (PR)

The Probabilistic Receiver is designed to minimize the number of errors in resolving the transmitted configuration. On receiving an amplitude, PR chooses the configuration that minimizes the probability of error in decoding that amplitude, in turn, maximizing the

probability of success in decoding the received amplitude. PR follows the Maximum a Posteriori (MAP) detection rule [**map˙book**] to minimize the bit error in decoding a configuration, given the received amplitude.

Observations on Optimality and Practicality of PR

The received amplitude is the sum of the magnitude of the transmitted configuration and the amplitude errors due to channel noise. The receiver observes this noisy signal and estimates the transmitted configuration using a priori estimates of the channel and the transmitter distributions. We use the probability of error in decoding the configuration as the metric to evaluate the performance of the receiver. We follow the logic of MAP detection rule that minimizes the expected symbol decoding error to define an optimum receiver [**map˙book**].

An optimum receiver is a receiver which minimizes the probability of error in resolving the transmitted configuration given the received amplitude.

Minimizing the probability of error on receiving a configuration $\hat{C}_i$ in turn maximizes the probability of success.

$$\Pr(\hat{C}_i \neq C_i) = 1 - \Pr(\hat{C}_i = C_i) \tag{4.2}$$

where C_i is the configuration transmitted. The receiver estimates $\hat{C}_i$ on receiving y. The receiver decision is considered a success if the estimated configuration was the actual transmitted configuration. On receiving an amplitude y, the probability of success in decoding the transmitted configuration is,

$$\Pr(\hat{C}_i = C_i, y) = \Pr(\hat{C}_i = C_i \mid y) \cdot \Pr(y) \tag{4.3}$$

Since $\Pr(y)$, the probability of receiving amplitude y, is a constant for a known source distribution and channel model, the optimum receiver will choose $\hat{C}_i$ such that it maximizes the conditional probability $\Pr(\hat{C}_i = C_i \mid y)$. Thus, the probability of success in choosing a

configuration $\hat{C}_i$ on receiving y is,

$$\Pr(\hat{C}_i \mid y) = \frac{\Pr(y \mid \hat{C}_i) \cdot \Pr(\hat{C}_i)}{\Pr(y)} \tag{4.4}$$

where $\Pr(\hat{C}_i)$, the probability of $\hat{C}_i$ being transmitted, is obtained from a priori estimates of the source distribution, $\Pr(y)$ is known for given amplitudes, and $\Pr(y \mid \hat{C}_i)$ from the a priori estimates of the channel transition probabilities. Thus, for every received amplitude y, the optimum receiver chooses a configuration that maximizes the probability of success with accurate a priori estimates of the source and the channel.

We design PR using the above MAP detection rule and iterate through all possible configurations and choose the most probable configuration which maximizes the overall probability of success given by,

$$\sum_{y=0}^{R_{\max}} \Pr(\hat{C}_i, y) = \sum_{y=0}^{R_{\max}} \Pr(\hat{C}_i \mid y) \cdot \Pr(y) \tag{4.5}$$

PR is therefore an optimum receiver that maximizes the probability of success in decoding the transmitted configuration for a received amplitude, with an accurate a priori estimate of the source distribution and the channel transition probabilities.

Though PR maximizes the probability of success in estimating the transmitted configuration, it is an idealized receiver that assumes accurate a priori estimates of the source and channel distributions. It is computationally complex to obtain an accurate estimate of these distributions. The computational complexity of the receiver to iterate through all possible configurations for each received amplitude increases exponentially with the number sources as the number of configurations increases exponentially with the number of sources.

To overcome these challenges, we propose Deterministic Receiver (DR), a practical, low-complexity, heuristic receiver design later in this section. PR maximizes the probability of success in decoding the configuration while DR focuses on individual bits, i.e.,

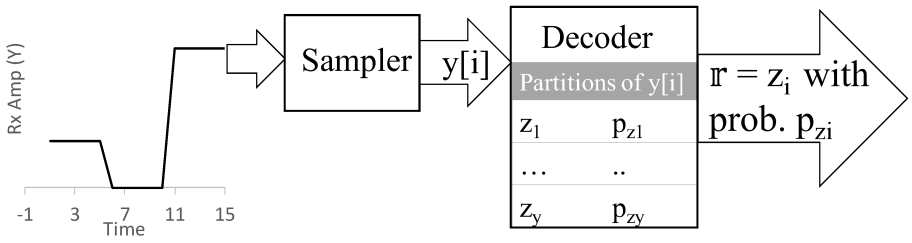


Figure 4.4: Probabilistic Receiver Illustration

PR minimizes the symbol error rate while DR focuses on reducing the bit error rate of individual sources.

We derive an upper bound on the expected number of successes using PR, which is then used to develop DR and the amplitude assignment algorithm in Section 4.10.

Probabilistic Receiver Architecture

Figure 4.4 is an illustration of the PR architecture. The *sampler* in this architecture is the *inverse model* derived from [16]. The sampler takes as input the response of bacteria and generates a time sequence of amplitude samples as received by the receiver bacteria. The *decoder* of PR takes a sampled signal $y[i]$ as input and generates a table of partitions of $y[i]$. z_1 to z_j are the partitions of $y[i]$. The decoder chooses a partition that maximizes the probability of success. The a priori channel transition probabilities are considered in generating the table of partitions for each received amplitude. If more than one partition can maximize the probability of success, one of them is chosen randomly as it will not affect the performance statistically.

Consider the example in Table 4.2. Each configuration has a magnitude and a received amplitude. For simplicity, we do not present channel transition probabilities in Table 4.2. In the presence of channel errors, the received amplitude for each configuration will be a range of amplitudes with a probability associated to each amplitude, unlike the single received amplitude shown here. Thus, in an ideal channel, on receiving an amplitude 3, PR chooses configuration $\{1, 1, 0\}$ with a probability $\frac{(1-p_t)(p_t)^2}{\Pr(y=3)}$.

It can be noted that the conditional probability is affected by two types of errors,

- channel error - channel induced errors alter the amplitudes received,
- collision error - when multiple sources collide in the channel, receiver receives the sum of amplitudes.

PR decodes C_i on receiving an amplitude y based on its probability of occurrence, i.e., $\Pr(C_i \mid y)$ is equal to $\frac{\Pr(C_i)}{\Pr(y)}$, if the channel is noise free, as shown in Table 4.3. With the knowledge of the transition probabilities of the channel, PR chooses a partition C_i as shown in Equation (4.4).

In the following section, we derive an upper bound on the expected number of successful address resolutions using PR. The number of successful address resolutions can be determined accurately for a known amplitude assignment. We derive the upper bound to compare different amplitude assignment mechanisms. There are $\binom{N}{R_{\text{max}}}$ ways of assigning R_{max} amplitudes to N sources. The upper bound is then used as a measure to evaluate the data-rate performance of different sets of addresses. In the derivation of the upper bound, we consider only the decoding error from collisions. The channel noise is a function of the amplitudes assigned, and therefore it becomes impossible to derive an upper bound in the presence of channel noise, without the knowledge of the amplitudes. In the implementation and evaluation of ADMA in nanoNS3, we introduce channel noise. As shown in Figure 4.4, PR maintains the list of partitions for each received amplitude. For a given p_t, the receiver determines the a priori probabilities of each partition. The decoder chooses one of the partitions with a probability equal to its probability of occurrence. An example of PR decoder table for $R_{\text{max}} = 3, N = 3, p_t = 0.5$ is shown in Table 4.3. Column 3 shows the conditional probabilities of each configuration given the received amplitude. On receiving amplitude 3, all five partitions have the equal probability of being chosen, and hence the conditional probability of choosing a partition on receiving amplitude 3 is 0.2. The probability of decoding a signal is proportional to that of its occurrence. A partition with a high probability of transmission is received with high probability. Let Y be the random variable representing the received amplitude and m_k be the bit sample transmitted by source u_k at

a given time and $\hat{m_k}$ be the bit sample the receiver decodes. Decoding is successful for source u_k if $m_k = \hat{m_k}$. The expected number of success, where success is the event of accurate address resolution, is given by

$$\mathbb{E}(\text{No. of succ}) = \sum_{y=1}^{R_{\max}} \mathbb{E}(\text{No. of succ.} \mid Y = y) \cdot \Pr(Y = y)$$

$$\mathbb{E}(\text{No. of succ.} \mid Y = y) = \sum_{k=1}^{N} \Pr(\hat{m_k} = m_k \mid Y = y) \tag{4.6}$$

where $\Pr(\hat{m_k} = m_k \mid Y = y)$ is the probability of success for source u_k on receiving y. We derive the probability of success for source u_k using PR below. As the sources are always backlogged, the message bit is either 1 or 0 and hence $m_k \in \{0, 1\}$. We further condition on $m_k = 1$ and $m_k = 0$. Let $p_y = \Pr(Y = y)$ be the probability of receiving amplitude y. Using Bayes's rule,

$$\begin{aligned}
&\Pr(\hat{m_k} = m_k, m_k = 1 \mid Y = y) \\
&\qquad = \frac{\Pr(\hat{m_k} = m_k, m_k = 1, Y = y)}{\Pr(Y = y)} \\
&\qquad = \frac{\Pr(\hat{m_k} = m_k \mid m_k = 1, Y = y) \cdot p_{yk^1}}{p_y}
\end{aligned}$$

where $p_{yk^1} = \Pr(m_k = 1, Y = y)$ is the probability of source u_k transmitting bit 1 and $Y = y$. A source always has a bit to transmit. Therefore, $p_{yk^1} + p_{yk^0} = p_y$

$$\Pr(\hat{m_k} = m_k, m_k = 0 \mid Y = y)$$

Table 4.3: Probabilistic Rx Example

Rx Amplitude	Partitions	Conditional Prob.
0	{0,0,0}	1
1	{1,0,0}	1
2	{0,1,0}	1
3	{0,0,1},{0,1,1},{1,0,1},{1,1,0},{1,1,1}	0.2

$$= \frac{\Pr(\hat{m_k} = m_k \mid m_k = 0, Y = y) \cdot p_{yk^0}}{p_y}$$
$$= \frac{\Pr(\hat{m_k} = m_k \mid m_k = 0, Y = y) \cdot (p_y - p_{yk^0})}{p_y}$$

Since PR chooses a partition with a probability equal to its conditional probability, on receiving an amplitude y given $m_k = 1$, u_k is successfully received, if the receiver chooses any one of the partitions such that $m_k = 1$.

$$\Pr(\hat{m_k} = m_k \mid m_k = 1, Y = y) = \frac{p_{yk^1}}{p_y} \tag{4.7}$$
$$\Pr(\hat{m_k} = m_k \mid m_k = 0, Y = y) = \frac{p_{yk^0}}{p_y} = \frac{p_y - p_{yk^1}}{p_y}$$
$$\Pr(\hat{m_k} = m_k, m_k = 1 \mid Y = y) = \frac{p_{yk^1} \cdot p_{yk^1}}{p_y \cdot p_y} \tag{4.8}$$
$$\Pr(\hat{m_k} = m_k, m_k = 0 \mid Y = y) = \frac{p_{yk^0} \cdot p_{yk^0}}{p_y \cdot p_y}$$

Substituting above probabilities, we derive,

$$\Pr(\hat{m_k} = m_k \mid Y = y) = \left(\frac{p_{yk^1}}{p_y}\right)^2 + \left(\frac{p_{yk^0}}{p_y}\right)^2 \tag{4.9}$$

The expected number of success depends on the probability of success of all sources and hence an upper bound on the probability of success per source will provide an upper bound on the expected number of success. Substituting Equation (4.9) in Equation (4.6),

$$\mathbb{E}(\text{No. of succ.} \mid Y = y) = \sum_{k=1}^{N} \left(\frac{p_{yk^1}}{p_y}\right)^2 + \left(\frac{p_{yk^0}}{p_y}\right)^2 \tag{4.10}$$

$$\mathbb{E}(\text{No. of succ.}) = \sum_{y=1}^{R_{\max}} \sum_{k=1}^{N} \Pr(\hat{m_k} = m_k \mid Y = y) \cdot p_y$$
$$\leq \sum_{y=1}^{R_{\max}} N \cdot \left(\frac{(p_{yk^1})^2}{p_y} + \frac{(p_y - p_{yk^1})^2}{p_y}\right) \tag{4.11}$$

From Equation (4.11), note that the expected number of success is maximized when $p_{yk^1} = p_y$. This supports the theorem that the number of successes is N only when the number of partitions is 1. Equation (4.11) can be further simplified as

$$\mathbb{E}(\text{No. of succ.}) \leq N \sum_{y=1}^{R_{\max}} \left(p_y - \frac{p_y - p_{yk^1}}{p_y} 2p_{yk^1}\right) \tag{4.12}$$

$p_{yk^1} \leq p_y$. When the equality does not hold we can write $p_{yk^1} = r \cdot p_y$ where $r < 1$

$$\mathbb{E}(\text{No. of succ.}) \leq N \sum_{y=1}^{R_{\max}} \{p_y \cdot (1 - 2r + 2r^2)\} \tag{4.13}$$

Average network throughput = $\frac{\text{Expected number of successes}}{\text{Pulse duration}}$. For each signal received, PR chooses one of the partitions with its probability of occurrence. For a set of integer amplitudes, the number of partitions increases exponentially [80] for large values of N. On receiving a signal, the receiver goes through the partitions of integers contributing $\mathcal{O}(e^N)$ to the time complexity; where N is the number of sources. This is repeated for each signal received and hence the overall time complexity is $\mathcal{O}(e^N\,|Z|)$, where $|Z|$ is the number of received signals. As the number of partitions increases exponentially, the space required to store all the partitions is given by $\mathcal{O}(e^N)$. The exponential time and space complexity limit the practical implementation of PR. In the following section, we propose a simpler receiver with reduced time and space complexity and an improved network throughput.

Algorithm 1 Deterministic Rx Implementation

$N \leftarrow$ `Number of sources`, $R(y) \leftarrow \{\}$
for $y := 0$ to R_{max} **do**
  for $k := 1$ to N **do**
    $p_{yk^1} \leftarrow \Pr\{\text{Source } k \text{ transmitting bit 1} \mid \text{y}\}$
    $p_{yk^0} \leftarrow \Pr\{\text{source } k \text{ transmitting bit 0}\}$
    if $p_{yk^0} \geq p_{yk^1}$ **then**
      $R(y) \leftarrow R(y), \{0\}$
    else
      $R(y) \leftarrow R(y), \{1\}$
    end if
  end for
end for

Deterministic Receiver (DR)

PR maximizes the probability of success with accurate estimates of source distribution and channel noise model. Also, the decoder complexity is exponential to the number of sources. To reduce the complexity of the receiver, we propose Deterministic Receiver (DR), a heuristic receiver design which chooses a pre-determined configuration on receiving an amplitude to reduce the bit error of each source independently.

Deterministic Receiver Architecture

For each received amplitude y, DR chooses a bit sample for individual sources independently. DR goes through all the partitions of the amplitude y and chooses the most probable bit for each source. This is pre-determined during receiver setup. In this work, we focus on minimizing decoding errors due to collisions in DR analysis. Knowledge of channel transition probabilities is incorporated by including channel noise in the estimation of the most probable bit. DR thus differs from PR in building and updating the decoder table. With the help of Algorithm 1, we explain the operation of DR.

Line 2 loops over all receivable amplitudes. For each amplitude received, DR compares p_{yk^1} and p_{yk^0} for each source, i.e., the probability of the source transmitting bit 1 and bit 0 given that amplitude y was received, is compared. In line 6, if $p_{yk^0} \geq p_{yk^1}$ for source u_k,

the receiver updates the received signal $R(y) = 0$ for source u_k. $R(y)$ is the vector of bits decoded by the receiver on receiving y. $R(y)$ is calculated once, and the receiver uses a constant-time lookup to decode. DR updates the vector of bits decoded for all amplitudes that can be received. DR maximizes the conditional probability of source u_k transmitting a bit given the received signal y. The maximization is performed a priori, and hence the time complexity is $\mathcal{O}(1)$. The bit vector calculated independently for each source may not be one of the partitions or configurations of the amplitude received. But this vector maximizes the number of successes for each source independently. We study the average performance of DR for different states of the receiver. To study the performance of DR, we use the same parameters as that of PR. The average number of successes using DR can be described as,

$$\mathbb{E}(\text{No. of succ.}) \leq N * \max_{1 \leq k \leq N} \max_{p_{y^1}, p_{y^0}} \leq N * \max_{1 \leq k \leq N} p_y * max(r, 1-r)$$

It can be observed that the bound on the probability of success reaches its maximum value of 1 at $r = 0$ or $r = 1$. We also proved in Theorem 1 that a practical system can achieve a probability of success of 1 only when $N \leq \lfloor \log_2(R_{\text{max}} + 1) \rfloor$. We derive a practical upper bound on the expected number of successes with DR, analytically.

Case 1: Chosen configuration is a partition If the chosen configuration is one of the partitions, then the expected number of successes when it is transmitted is N. When the chosen configuration is transmitted, it will be decoded without error. All the partitions that differ from the chosen configuration by one bit will be received with $N - 1$ successes and so on.

Table 4.4: DR: Upper bound estimation

Max No. of successes	Case 1	Case 2
N	1	0
$N-1$	$N-1$	1
$N-2$	$\leq \binom{N}{2}$	$\leq \binom{N}{2}$
$N-3$	$\leq \binom{N}{3}$	$\leq \binom{N}{3}$
..	..	..
1	$\leq \binom{N}{N-1}$	$\leq \binom{N}{N-1}$
0	0	1

Algorithm 2 Deterministic Receiver Upper bound

$R \leftarrow R_{\text{max}}$
$N \leftarrow$ Number of sources
$\mathbb{E} \leftarrow 0$, Expected number of successes
for $i := 1$ to R **do**
    $\mathbb{E} \leftarrow \mathbb{E} + C[0:i] \cdot N, \hat{C} \leftarrow C[i : \text{end}]$
    $\mathbb{E} \leftarrow \mathbb{E} + \hat{C}[0:(R-i)] \cdot (N-1)$
    $\hat{C} \leftarrow C[(R-i) : \text{end}]$
    $\mathbb{E} \leftarrow \mathbb{E} + C[0 : \binom{N}{1}] \cdot (N-1)$
    $\hat{C} \leftarrow C[R + \binom{N}{1} : \text{end}]$
    $\mathbb{E} \leftarrow \mathbb{E} + \hat{C}[0 : \binom{N}{2}] \cdot (N-2)$
    $\hat{C} \leftarrow C[R + \binom{N}{1} + \binom{N}{2} : \text{end}]$
    $\mathbb{E} \leftarrow \mathbb{E} + \hat{C}[0 : (R-i) \cdot \binom{N}{2}] \cdot (N-2)$
    $\hat{C} \leftarrow C[R + \binom{N}{1} + \binom{N}{2} + (R-i) \cdot \binom{N}{2} : \text{end}]$
    $i_{start} \leftarrow R + \binom{N}{1} + \binom{N}{2} + (R-i) * \binom{N}{2}$
    $i_{\text{end}} \leftarrow i_{start} + \binom{N}{3} + \binom{N}{2} + (R-i) \cdot \binom{N}{3}$
    for $j = 3$ to N **do**
        $\mathbb{E} \leftarrow \mathbb{E} + C[i_{start} : i_{\text{end}}], i_{start} \leftarrow i_{\text{end}}$
        $i_{\text{end}} \leftarrow i_{start} + i \cdot \binom{N}{j} + (R-i) \cdot \binom{N}{j}$
    end for
end for

All configurations that sum to y will be different from the chosen one by at least one bit. No two configurations that differ by one bit can add to the same integer. For example, a configuration 1101 and 1100 differs by one bit. Let y_1 be the sum of configuration 110. $y_1 + a_4$ cannot be equal to y_1. Thus, if the chosen configuration is one of the partitions,

no partition of that sum will have $N-1$ success. Configurations that map to R_{max} is an exception. Since all configurations that sum to R_{max} and greater are mapped to R_{max}, two configurations that differ by one bit can add to R_{max}. For example, configurations 001 and 011 are mapped to 3 in Table 4.4. Therefore, at most $N-1$ configurations that differ by one bit from chosen configuration and mapped to R_{max} can have $N-1$ successes.

Following the same argument, no partition of a sum can differ from the chosen one by two bits if the chosen configuration is a partition. For example, 1101 and 1110. Let sum of amplitude 11 be y_2. $y_2 + a_4$ cannot be equal to $y_2 + a_3$ as all amplitudes are distinct. Therefore, if the chosen one is one of the partitions, no configuration that sum to X will have $\leq N-3$ successes. Configurations summing to R_{max} are an exception. Up to $\binom{N}{2}$ configurations that differ from the chosen configuration that add to R_{max} can have $N-2$ successes. There can be up to $\binom{N}{3}$ partitions with a Hamming distance of 3 between themselves and the chosen configuration and up to $\binom{N}{4}$ partitions with a Hamming distance of 4 from the chosen partition and so on. Column 1 in Figure 4.4 shows the maximum number of successes and Column 2 defines the maximum number of configurations that can achieve these successes in Case 1.

Case 2: Chosen configuration is not a partition If the chosen configuration is not one of the partitions, none of the partitions of that integer can have N success. At most, one partition can have $N-1$ success. At most, one partition has a Hamming distance of one from the decoded configuration. Let us assume there are two partitions with a Hamming distance of one between them and the decoded configuration. Then the Hamming distance between these partitions is at most 2. Since two partitions of an integer differ by at least 3 bits, this is not possible. Thus, if the chosen configuration is not one of the partitions, up to one configuration can have $N-1$ success. There can be up to $\binom{N}{2}$ partitions that have a Hamming distance of 2 from the chosen configuration. Similarly, there can be up to $\binom{N}{3}$ partitions that are 3 Hamming distance away from the chosen configuration. Columns 2 and 3 in Figure 4.4 summarizes Case 1 and 2 respectively. There are R_{max} distinct integers

receiver can receive. For some amplitudes, the receiver can choose one of its partitions; and for others, the receiver chooses a configuration that is not a partition. We use Algorithm 2 to find an upper bound using DR. C is the sorted array of the probability of each configuration being transmitted in descending order and $\mathbb{E}$ is the expected number of success on receiving an amplitude sample. Given N and p_t, there are $\binom{N}{k}$ configurations with k ones and $N-k$ zeros, where k varies from 0 to N. Lines 4 loops over i where i varies from 1 to R. i represents the number of received amplitudes whose decoded configuration is one of the partitions. $R-i$ amplitudes have decoded configurations that are not their partition. From the above table, we know that only those decoded configurations that are partitions of the amplitude can have N success. Thus, in the *for loop* in Algorithm 2, up to i configurations can have N success. To calculate an upper bound, we assume that the highest probable i configurations can have N success. Also, one configuration per received amplitude can have $N-1$ success, if the decoded configuration is not its partition. Thus, up to $R-i$ configurations can have $N-1$ success. All amplitudes greater than R_{max} is received as R_{max} , and hence the highest receivable amplitude can have up to N partitions with $N-1$ success and similarly up to $\binom{N}{2}$ configurations with $N-2$ successes. Lines 16 to 19 loop through remaining configurations assigning $N-j$ successes to $\binom{N}{j} \cdot N$ configurations. Maximum E over all i gives the upper bound on the expected number of success.

Amplitude Assignment Algorithm

As discussed above, the two factors that impact Collision Resolution Error (CoRE) are the receiver design and the source addresses. In this section, we present a heuristic algorithm that chooses an amplitude sequence that minimizes CoRE based on the insights we gained from the derivation of the probability of success in Section 4.7. Following Theorem 1, the maximum number of sources that can be accommodated with zero CoRE is $\lfloor \log_2(R_{\text{max}} + 1) \rfloor$. Since the addresses are distinct, the maximum number of sources is limited by R_{max}. As the number of sources increases, the required R_{max} to achieve zero CoRE increases

exponentially. In this section, we propose an algorithm to choose a set of amplitudes that minimizes CoRE, for a fixed N and $R_{\max}$. From the probability of success derivation for receiver design in Section 4.7, we obtain the following insights, which is then used to propose four amplitude sequences.

Insight 1. *The higher the number of partitions, the lower is the probability of success on receiving the sum.*

Insight 2. *At a low probability of transmitting a* bit 1*, the number of colliding signals is* ≤ 2 *with high probability.*

When the number of collisions is less than or equal to two, an address sequence that can recover from two collisions will have a high probability of success. We define *shifted (natural) sequence* as a sequence of integers with the first element shifted by $\frac{R_{\max}-1}{2}$, $S = \left\{\frac{R_{\max}-1}{2}, \frac{R_{\max}-1}{2}+1, \ldots, R_{\max}-1\right\}$. A maximum of $\frac{R_{\max}}{2}$ sources can be supported using this sequence. No element in the shifted sequence can be written as a sum of any two elements. $a_i + a_j \neq a_k$ where $a_i, a_j, a_k \in S$ and $i \neq j \neq k$. Therefore, when one source transmits a bit 1, it will be received without error. When two sources transmit bit 1 and collide, the number of partitions of received amplitude is reduced, and therefore the error is smaller compared to other address sequences since an error in one address will not affect others, making it suitable for scenarios with a low probability of bit 1 collisions. A shifted natural sequence allows $\frac{R_{\max}}{2}$ sources. If $N \geq \frac{R_{\max}}{2}$, we extrapolate the shifted sequence as $S = \{R_{\max} - N_u - 1, R_{\max} - N_u - 2, ...R_{\max} - 1, R_{\max}\}$. The extrapolated sequence does not hold the property of a shifted sequence. The number of elements that can be written as sum of two other elements is small in this sequence since its elements are decreasing integers from $R_{\max}$.

Insight 3. *At a high probability of transmitting a bit 1, the number of colliding signals is* $\geq N_{max}$ *with high probability.*

Let $N_{\max}$ be the number of sources such that sum of amplitudes up to $N_{\max}$ is less than equal to $R_{\max}$. When more than $N_{\max}$ sources collide, a *natural sequence*, which is the sequence of integers beginning from 1 till N, will have a high probability of success. When the number of sources transmitting bit 1 at a given time increases, the probability of receiving configurations that sum to $\geq R_{\max}$ increases. All sums greater than $R_{\max}$ are received as $R_{\max}$. On receiving a sum $R_{\max}$, the receiver chooses one of the configurations with sum $\geq R_{\max}$. The probability of choosing a particular configuration is $\frac{1}{N(x \geq R_{\max})}$, where $N(x \geq R_{\max})$ is the number of configurations that sum up to $R_{\max}$ and more. The fewer the number of configurations with sum $\geq R_{\max}$, the higher the probability of success. Amplitudes that can minimize $N(x \geq R_{\max})$ will minimize CoRE. A natural sequence has the maximum number of configurations $< R_{\max}$, and hence minimum $N(x \geq R_{\max})$.

Algorithm 3 Address Allocation

$R_{\max} \leftarrow$ Maximum Receivable amplitude
$N_s \leftarrow$ Number of simultaneous sources
$N_u \leftarrow$ Total number of sources
$S \leftarrow$ Set of addresses
if $\Pr(N_s \leq 2) > \Pr(N_s > 2)$ and $N_u \leq \frac{R_{\max}}{2}$ **then**
    $S \leftarrow \{\frac{R_{\max}-1}{2}, \frac{R_{\max}-1}{2} + 1, \ldots, R_{\max} - 1\}$
else if $\Pr(N_s \leq 2) > \Pr(N_s > 2)$ and $N_u > \frac{R_{\max}}{2}$ **then**
    $S \leftarrow \{R_{\max} - N_u, R_{\max} - N_u - 1, \ldots, R_{\max} - 1\}$
else if $\Pr(N_s \leq N_{\log}) > \Pr(N_s > N_{\log})$ **then**
    $S \leftarrow \{2^0, 2^1, \ldots, 2^{N_{\log}}, R_{\max}, R_{\max} - 1, \ldots\}$
else if $\Pr(N_s \geq N_{\max}) > \Pr(N_s < N_{\max})$ **then**
    $S \leftarrow \{1, 2, 3, \ldots, N_u\}$
else $S \leftarrow \{2^0, 2^1, \ldots, 2^{N_{\log}}, R_{\max}, R_{\max} - 1, \ldots\}$
end if

Insight 4. *At an intermediate probability of transmitting a bit 1, the number of colliding signals is $\leq N_{\log}$ with high probability.*

At intermediate values of p_t, the per-source probability of transmitting a bit 1, up to $N_{\log}$ signals collide. A sequence that has the maximum number of sums with unique configurations with up to N elements will have a high probability of success. *Extrapolated binary*

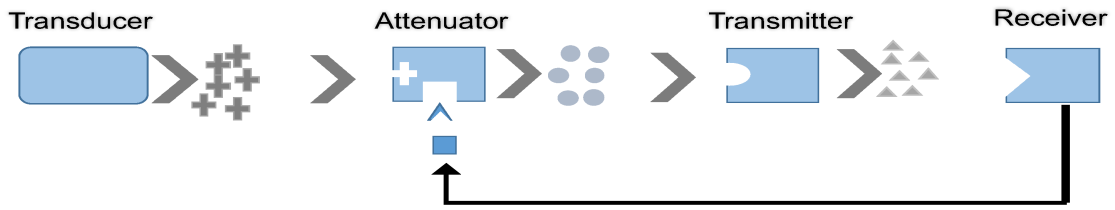


Figure 4.5: Practical Implementation: Illustration

sequence $S = \{2^0, 2^1, 2^2, \ldots N_{\log}, R_{\max} - N - N_{\log}, R_{\max} - N - N_{\log} - 1, \ldots, R_{\max}\}$ combines a binary sequence and a shifted sequence. $N_{\log} = \lfloor \log_2(R_{\max} + 1 \rfloor)$ is the number of sources a binary sequence can support. A binary sequence is an optimum solution to achieve zero CoRE. An extrapolated binary sequence utilizes the maximum number of distinct sums when using a binary sequence. These distinct sums are obtained from the binary elements. The rest of the elements are integers decreasing from $R_{\max}$, which reduces the number of overlapping sums. As the number of collisions approaches $N_{\log}$, the distinct sums contributed by the binary elements will improve the probability of success. Algorithm 3 summarizes the insights gained from the sequences observed and the probability of success derivation.

Amplitude assignment : Practical Implementation

We presented an optimum receiver design and an algorithm to choose an amplitude sequence that minimizes CoRE. In this section, we discuss network architecture and a practical bacterial communication system that considers practical amplitude assignment. We describe how a receiver can assign amplitudes to the sources based on its $R_{\max}$. To the best of our knowledge, no existing work provides a practical transmitter design that can transmit different amplitudes using only components built from bacterial populations. In Figure 4.5, we present a circuit design to implement a bacterial transmitter for ADMA. As shown in Figure 4.5, each source consists of three major components viz., a transducer, an attenuator and a transmitter. Each *source* in Figure 4.1 is composed of the above three components. In a sensing network, the transducer is genetically engineered, based on the application, to sense a specific signal and convert to another signal, as it is now possible

to design networks that utilize multiple signal molecules with little cross-talk [81]. The transducer emits one quorum sensing signal (AIP for example, crosses) at a concentration proportional to the inducer (IPTG) concentration. The AIP thus generated is attenuated by the amplifier/attenuator, which produces a second signal (AI-2 for example, circles). The amount of attenuation can be controlled externally using feedback. This attenuated AI-2 signal is the input to the transmitter which in turn emits a distinct signal (C6-HSL for example, triangles) to be transmitted to the receiver through a microfluidic channel. The attenuator modifies the transmitted signal concentration, and therefore is the address allocator in the architecture. Assuming a feedback path from the receiver to the attenuator, the receiver assigns the amplitude to each source of bacterial population by modifying the attenuator. The attenuator can be a bacterial population that emits AI-2 in a manner so that the concentration of AI-2 emitted is controlled by an external trigger (squares) to the attenuator population or additional signal (such as AIP itself) from the receiver. Such a trigger is provided by the receiver (or a centralized server) to each source based on the amplitude assigned to the source. Thus, the receiver (or the server) controls the amplitude/concentration of the signal transmitted to the receiver which in turn is used to identify the source on receiving a signal. It can be noted that the concentration of molecules generated by all the sources is the same. Thus, ADMA is also an energy fair mechanism.

Performance Evaluation

We evaluate *ADMA* in progressive steps.

1. We evaluate the performance of amplitude assignment algorithm and the receiver designs, using BCS (Bacterial Communication Simulator), a custom-built Python-based simulator in idealized channel conditions, i.e.., assuming zero sampling and demodulation error at the receiver.

2. We introduce channel errors in BCS and analyze the performance of ADMA with chan-

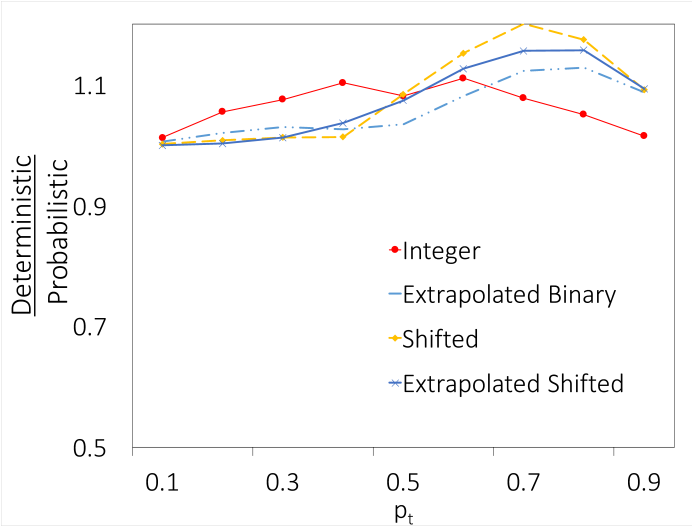


Figure 4.6: Data-rate of Deterministic vs. Probabilistic Receiver $N = 5, R_{\max} = 15$

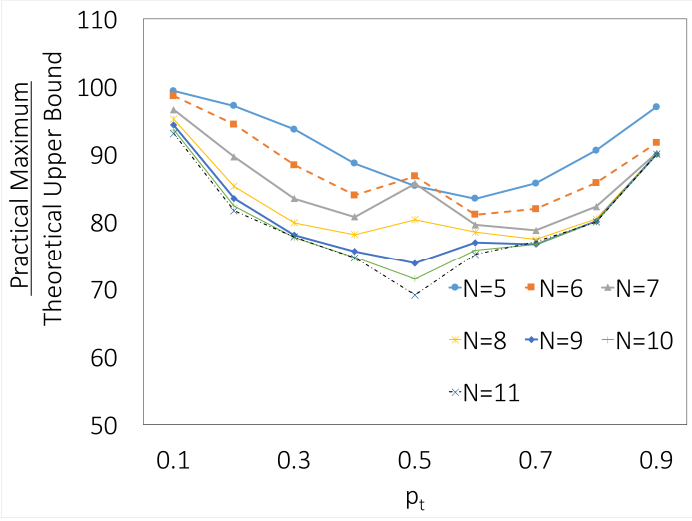


Figure 4.7: Theoretical vs. Practical Upper Bound

nel errors.

3. We implement the response of a receiver bacteria located in a microfluidic chip derived in [16] in nanoNS3 [82], the bacterial molecular communication simulator developed on top of NS3 simulates source error, channel error, and sampling and demodulation error at the receiver. We implement *amplitude assignment* algorithm and *deterministic receiver (DR)* in nanoNS3 and evaluate the performance with errors. We also show using experimental results and nanoNS3, that by using the inverse of the receiver model, receiver response can be demodulated.

nanoNS3

There are several existing works focusing on simulating Molecular Communication (MC) [83, 84, 85, 86, 87, 88, 89, 90]. Table 4.5 lists the properties of some of the aforementioned simulators focusing on the MC network. These approaches validate their respective simulators using numerical analysis or purely simplified theoretical models. Thus, the simulators are not verified against real-life behaviors. We discuss in detail some of the MC network simulators. NanoNS is built on top of Network Simulator 2 (ns-2), and it provides various nanoscale communication paradigms based on a diffusive MC channel [83]. This work only presents the details of the channel layer, and it simulates the diffusion and reception process using a single equation, which may not be accurate in the practical situation. This work simulates MC using molecules based approach, which is time-consuming as the molecule scales (for practical cases, the size of molecules is immense). Also, this work is based on ns-2 which is computationally inefficient with regards to memory usage and CPU utilization. Currently, ns-2 is not actively maintained, and the most recent version of ns-2 was released in 2011. dMCS developed in [89] proposes a simulation framework for the general case of diffusion-based MC, and it is developed using a customized simulator. Using customized simulator is likely to lose the advantages of dedicated network simulators like ns-3 (e.g. scalability and computational efficiency). Again, [89] is also modeling

Table 4.5: Simulators Comparison

Features/Simulator	NanoNS	N3Sim	Nano-Sim	nanoNS3
Physical Layer/Channel Model	Diffusive channel, Gillespie model for stochastic reaction process	Baraff's algorithm simulating collision	Spectrum channel model	Bacterial receiver model and channel loss model
Protocols Implemented	Molecular node emitting molecules	Emitter types to generate molecules pulse trains	Selective and Random routing, Transparent-MAC, Smart-MAC, Time spread OOK	OOK, Source addressing, Error analysis and transfer rate analysis model, ns-3 application layer
Validation	Numerical analysis of models used	N/A	N/A	Experimental evaluation
Compatibility of Network Simulator	ns-2	N/A	ns-3	ns-3
Programming language used	Tcl, C++	Java	C++	C++
Modularity of Network Architecture	Higher layer protocols of ns-2 can be integrated with the physical layer	Does not have provision to implement higher layer protocols	Higher layer protocols of ns-3	Higher layer protocols of ns-3

MC network using molecules based approach, which incurs large time complexity as the number of molecules scales and no higher layer protocol is implemented in this work.

In [84], N3Sim is developed based on the diffusion propagation channel to model MC networks. Similar to [89], [84] is built on a customized simulator and those network layers higher than PHY is absent in this work. N3Sim allows us to configure the network using a configuration file on the front end making it easy to use the simulator. N3Sim does not follow layered architecture in simulating the network. It focuses primarily on the physical layer diffusive channel. Therefore, we cannot use N3Sim to simulate, compare and analyze the performance of MC network for upper layer algorithms. Nano-Sim developed in [85] is also built on top of ns-3, and it provides functions to model Electromagnetic (EM) wave based nanonetworks. Similar to our work, Nano-Sim utilizes the framework and advantages of ns-3 to build EM-based nano simulator. The transmission/reception scheme in Nano-Sim is orthogonal to our work in this paper. Thus, it is feasible to combine *nanoNS3* with Nano-Sim, since they are both implemented atop ns-3. Other than aforementioned MC simulators, [88] proposes a simulation framework that is adaptable to any kind of nano bearer and the simulator is also validated using experimental analysis in [91], but it is developed using a customized simulator. Thus, it is likely to lose the advantages of dedicated network simulators. To the best of our knowledge, *nanoNS3* is the first BMC network simulator validated using experimental analysis that achieves a demodulation accuracy greater than **92.5%**.

Network Architecture :

nanoNS3 is developed atop ns-3 [92]. ns-3 is a discrete event, open source and widely used network simulator for internet systems, targeted primarily for research and educational use (ns-3 is developed in C++ and python). ns-3 is developed based on modules, and each individual module represents a protocol (e.g. AODV), a technology (e.g. WiFi) or an attribute of networks (e.g. mobility). It enables the easy and convenient upgrade of source code and triggers the ease of extensibility in ns-3 by this modular implementation method. ns-3 is

actively maintained and it is free software and licensed under GNU GPLv2 license. ns-3 has the best overall performance compared with other popular network simulators [93]. E.g. ns-3 has the least memory usage for large-scale network simulations compared with ns-2, OMNeT++, JiST and SimPy. Implementing *nanoNS3* in ns-3 has the following major advantages: 1) open sourced availability and ease of implementation for new algorithms, 2) high computational efficiency for large-scale networks, and 3) supporting tools from ns-3 can be utilized directly (e.g. ns-3 logging and tracing systems).

nanoNS3 Network Architecture

The high-level structure of *nanoNS3* is shown in Fig. 4.8. The name of seven important classes with the structure of the corresponding network layers are given in Fig. 4.8. The functionality of each class is discussed briefly below:

- *NanoNetDevice*: It is similar to the Network Interface Card (NIC), and it can support different nano communication technologies (e.g. diffusive or EM wave based nano communication schemes) and corresponding protocols (e.g. amplitude addressing).

- *NanoNode*: It can be regarded as the physical device, and different *NanoNetDevices* can be integrated with *NanoNode* to provide corresponding communication technologies and protocols to enable *NanoNode* to communicate with each other.

- *PacketSocket*: This class is a simple and original ns-3 application class, which does not use IP addresses. It is used to set up user defined applications for nano communications by controlling application-related parameters, e.g., packet arriving interval, the number of maximal transmission packets and packet size.

- *NanoRouting*: This class manages message forwarding by each *NanoNode*.

- *NanoMAC*: This class manages channel access of different *NanoNodes*, and it also manages MAC layer addressing mechanism.

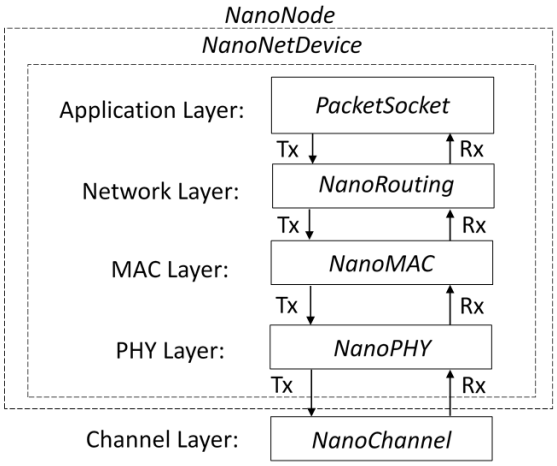


Figure 4.8: nanoNS3 Architecture

- *NanoPHY*: This class is used to simulate the process of transmitters and receivers to transmit and receive the nano signals. The corresponding functionality of this class includes modulation, demodulation, error analysis and transfer rate analysis, and receiver response.

- *NanoChannel*: This class is used to set up channel conditions, and then the channel loss can be calculated to simulate how the transmitted signals are propagated and attenuated in the corresponding microfluidic channel.

Specifically, packet client, packet server, and packet address are three classes related to packet socket class. The packet client class defines how packets are transmitted from Tx application layer, and the packet server class defines how packets are received from Rx application layer. The packet address class defines how addresses of different nodes are set up. We integrate the ns-3 original application class into *nanoNS3* , in order to make it more convenient to integrate other original ns-3 classes with *nanoNS3* , e.g. transport layer protocol. In ns-3, transport layer protocol is integrated with socket, so we integrate packet socket into *nanoNS3* to make it possible for transport layer extension. The parameters for each aforementioned class can be customized by users.

Idealized network conditions

Unless otherwise mentioned, we make the following assumptions in the implementation and evaluation of ADMA in BCS. Each source transmits bit 1 with a probability p_t and has an uninterrupted supply of data to transmit. Each data point in the results presented is averaged over 100 simulations. Before evaluating ADMA, we present the tightness of upper bound derived in Section 4.7.

Upper bound tightness:

In Algorithm 2, we derive an upper bound on the expected number of success, assuming that the highest probability configuration achieves the highest number of success without considering the amplitudes assigned. In practice, configurations are not independent of the amplitudes assigned and the probability of success depends on the choice of amplitudes and the receiver. We evaluate the theoretical upper bound tightness by performing an exhaustive search on all possible address assignment and determine the practical upper bound. Given R_{max}, N, there are $\binom{R_{\text{max}}}{N}$ possible address sequences; N addresses chosen from $[1, 2, 3, \ldots, R_{\text{max}}]$. The average expected number of success is calculated for each address sequence possible and compared to the theoretical upper bound. We limit our exhaustive search to a maximum of $R_{\text{max}} = 15$ as $\binom{R_{\text{max}}}{N}$ increases as the factorial of R_{max} increases and an exhaustive search is computationally not feasible. Figure 4.7 shows the ratio of practical bound to that of theoretical bound in y-axis as a function of p_t, for $R_{\text{max}} = 15$ and number of sources N ranging from 5 to 11. At smaller values of N, the probability of fewer "bit 1" collisions is higher, i.e., the probability of collision of $\frac{N}{2}$ or fewer sources is higher than the collision of more than $\frac{N}{2}$ sources and the probability of receiver saturation is small. As N increases, the probability of collision of $\frac{N}{2}$ or more sources increases, further increasing the probability of receiver saturation, which in turn decreases the effective throughput. For $R_{\text{max}} = 15, N = 5$, the practical bound is 99.3% of the theoretical bound at $p_t = 0.1$ and 91.2% on an average. Theoretical upper bound, on an average is 95.8% at

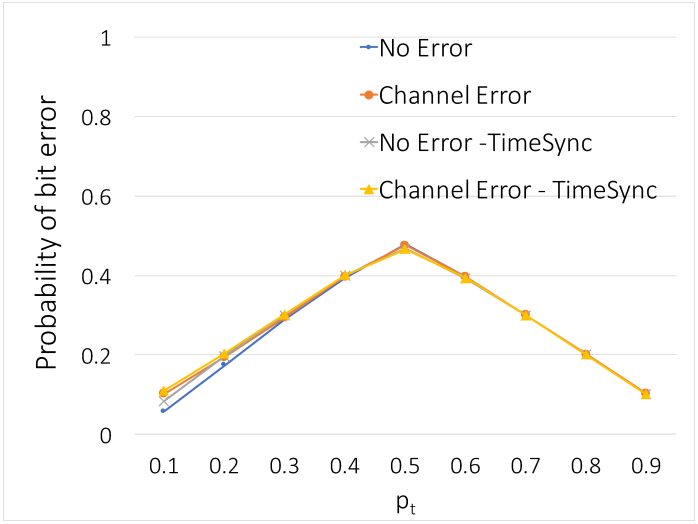


Figure 4.9: Bit error rate performance of ADMA, load aware DR,$R_{\max} = 30$, $N = 14$

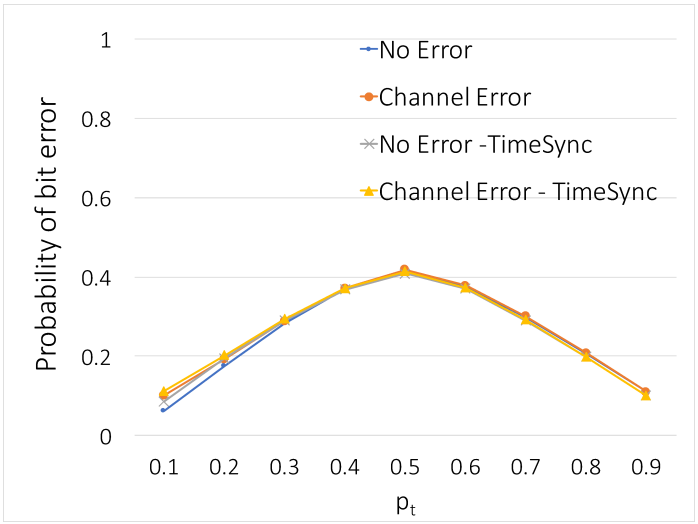


Figure 4.10: Bit error rate performance of ADMA, load aware DR,$R_{\max} = 70$, $N = 15$

$p_t = 0.1$, 78.9% at $p_t = 0.5$ and 91.3% at $p_t = 0.9$ for $R_{\text{max}} = 15$ and N varying from 5 to 11. At very low and very high p_t, the theoretical bound is close to 90% of the practical upper bound.

Load Aware Receiver

From Equation (1), we find that the parameters influencing CoRE are, 1. receiver design, 2. per-user probability of transmitting bit 1, p_t, 3. maximum receivable amplitude R_{max}, and 4. number of sources N. We evaluate the performance of ADMA by varying p_t, N and R_{max}. In Figure 4.6, the throughput performance of PR is compared against that of DR for $N = 5, R_{\text{max}} = 15$ under idealized conditions. The throughput achieved using DR outperforms that of PR for different sequences, which is attributed to the objective of each receiver design. The goal of DR is to reduce per-user bit error and increase the average per-user throughput, whereas PR maximizes the joint throughput of the network. We evaluate the performance of ADMA using DR in the rest of the section due to its low time complexity in implementation. We use bit error rate as the metric to evaluate ADMA for different sets of R_{max}, N and p_t, which is then used to calculate per-user throughput. We calculate the expected number of successful bits received per user from bit error and average over total time taken to calculate the average throughput. Figures 4.9 and 4.10 shows the bit error rate performance of ADMA for increasing values of p_t when receiver is aware of p_t. Four curves in each graph plots the bit error rate performance of ADMA in the presence and absence of channel error, and presence and absence of time synchronization. As discussed in Section 4.7, the receivers are designed to decode samples and do not require any time synchronization. Channel error is introduced by adding up to 20% amplitude error to 10% of the bits transmitted by each source. Figures 4.9 and 4.10 shows that ADMA is robust to asynchronous transmissions even in the presence of channel error. When integer amplitudes are assigned, DR chooses a configuration with the most probable bit for each source. The configuration chosen by DR for received amplitude y differs only by few sources for amplitudes close to y, making ADMA robust to channel error. For $R_{\text{max}} = 30, N = 14$,

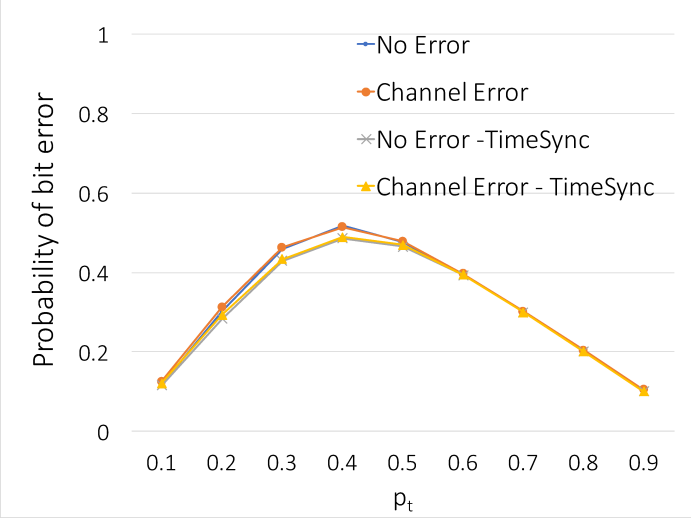


Figure 4.11: Bit error rate performance of ADMA, load unaware DR, $R_{\max} = 30$, $N = 14$

the throughput performance of ADMA is 92.5% of the upper bound at $p_t = 0.1$ and on an average 80.5% of the upper bound. For $R_{\max} = 70, N = 15$, the throughput performance of ADMA is 92.4% of the upper bound at $p_t = 0.1$ and on an average 82.3% of the upper bound. As shown in Figure 4.7, the practical bound on an average is close to 90% of the upper bound. Extrapolating this result to the performance at $R_{\max} = 70, N = 15$ and $R_{\max} = 30, N = 14$, the average performance of ADMA is close to 90% of the absolute maximum.

Load Unaware Receiver

In the above results, we assume that the receiver is aware of the load distribution (p_t). It may not be possible in a practical system to estimate p_t accurately. We propose a load unaware receiver that updates the decoding table for $p_t = 0.5$, when p_t is unknown. In a load unaware scenario, ADMA always chooses the integer address, i.e., $\{1, 2, 3, \ldots\}$. Figures 4.11 and 4.12 shows that the load unaware receiver does not affect the bit error rate performance of ADMA. For $R_{\max} = 30, N = 14$, the throughput performance of ADMA is 88.5% of the upper bound at $p_t = 0.1$ and 76.1% of the upper bound on an average. For $R_{\max} = 70, N = 15$, the throughput performance of ADMA is 91% of the upper bound at $p_t = 0.1$ and 81.4% of the upper bound on an average. We observed that the integer se-

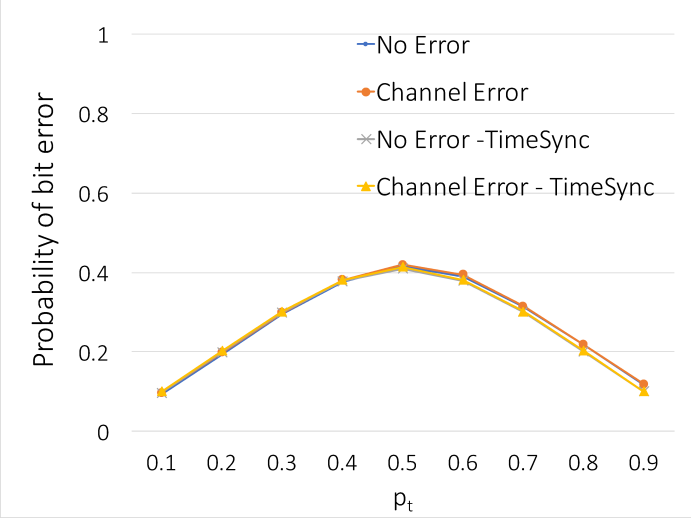


Figure 4.12: Bit error rate performance of ADMA, load unaware DR, $R_{\text{max}} = 70$, $N = 15$

quence has the best performance on an average when the receiver decodes using $p_t = 0.5$, simplifying amplitude assignment algorithm. In Figure 4.13 and 4.14, we plot the average number of successful frames using only integer sequence as a function of p_t for two sets of R_{max} and N. On average, the performance of load unaware DR using integer sequence remains within 95% of the load aware DR. Also, we note that the performance of integer sequence is within 99% of the algorithm performance on average.

ADMA with Bacteria Receiver in Microfluidic Channel

So far, we evaluated the performance of ADMA assuming zero sampling and demodulation error. Here, we implement and evaluate ADMA in nanoNS3 to simulate a practical bacterial communication system with source errors, channel errors and sampling and demodulation errors. We also show using experimental results and the receiver response model of [16] in nanoNS3 that the inverse of the receiver model can be used to demodulate the received signal with high accuracy. We implement the bacterial receiver response derived in [16] that models the performance of receiver bacteria in a microfluidic chip. The receiver model outputs the GFP response of the bacteria in the chamber for a given input. nanoNS3 generates a train of rectangular pulses for different sources and sums the signals

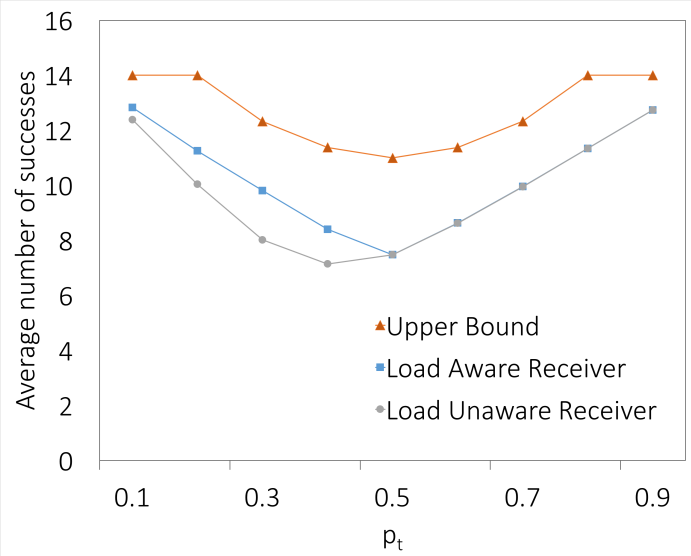


Figure 4.13: Performance of Interger sequence : N =14, $R_{max} = 30$

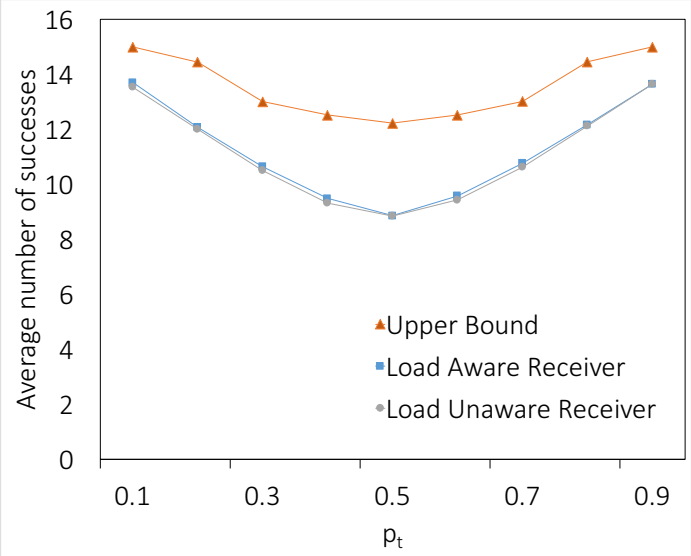


Figure 4.14: Performance of Interger sequence : N =15, $R_{max} = 70$

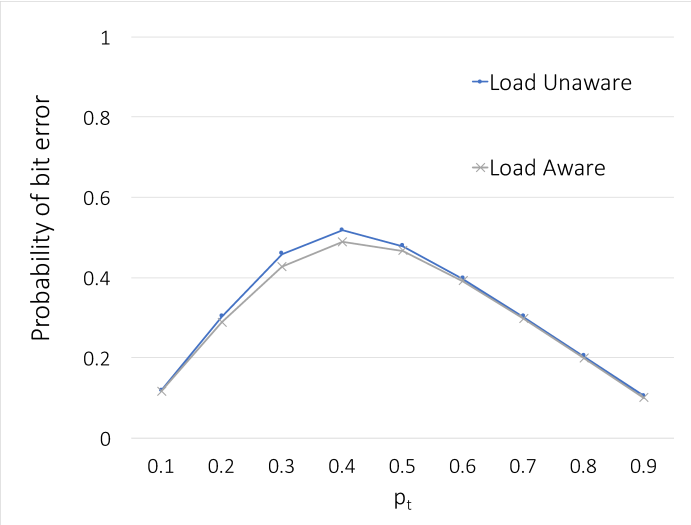


Figure 4.15: Bit error rate performance of ADMA in nanoNS3, $R_{\max} = 30$, $N = 14$

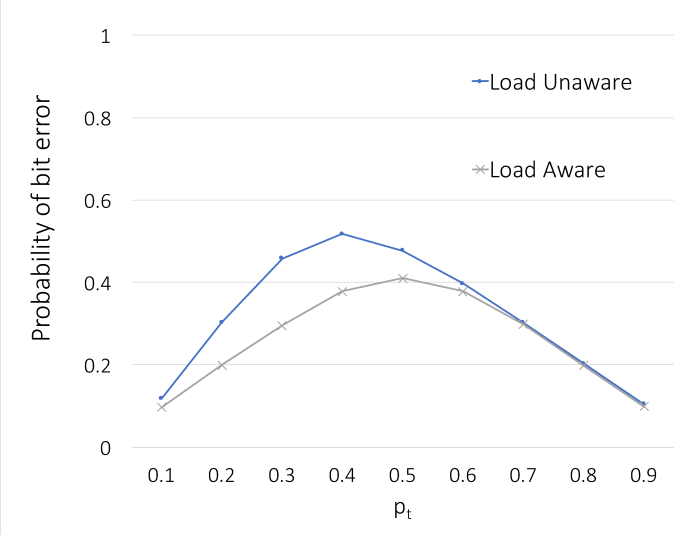


Figure 4.16: Bit error rate performance of ADMA in nanoNS3, $R_{\max} = 70$, $N = 15$

in the channel in time-domain. A channel attenuation model that attenuates the amplitude of the signal in the channel introduces a 20% amplitude attenuation per source. The cumulative signal (from multiple sources) is input to the receiver model, which generates the response of the receiver for the corresponding input. nanoNS3 builds an *inverse* of the model numerically, taking GFP response as input and outputs an estimate of the molecular signal. The output of *inverse model* is a time sequence of an estimated amplitude sample. Sampling and demodulation error is introduced at the receiver by varying "k parameters" of the inverse model which define different rate constants in the receiver response and varying them introduces random errors at the receiver. Each source is assigned a bit duration of 50 min and sampled every 10 min generating five samples. The samples from the *inverse model* are divided into blocks of 5 (5 samples per bit) and input to the decoder that outputs bit 1 if the average amplitude of a block of samples is greater than the average, and bit 0 if below. For example, a time sequence output of DR block for user u_1 with amplitude a_1, a_1,a_1,a_1,0,a_1, is decoded as bit 1 and 0,0,0,a_1,a_1 is decoded as bit 0.

nanoNS3 thus uses *forward response* model to generate GFP response of receiver bacteria and the *inverse model* to demodulate. The results from nanoNS3 thus represents a practical system by considering non-ideal conditions at the source, channel, demodulator and the receiver. Figures 4.15 and 4.16 show the bit error rate performance of ADMA in nanoNS3 for $R_{\text{max}} = 30, N = 14$ and $R_{\text{max}} = 70, N = 15$. The load unaware and load aware results are very close to each other, as actual input distribution $p_t \geq 0.5$. This is due to receiver saturation at R_{max}. All amplitudes greater than R_{max} are received as R_{max}. For $p_t \geq 0.5$, the probability of receiving R_{max} or higher is high, i.e., the receiver observes R_{max} with high probability and the number of partitions mapped to R_{max} is much higher than other received amplitudes. Therefore, even with $p_t \geq 0.5$, the probability of bit 1 and bit 0 is close to 0.5 on receiving R_{max}. For $R_{\text{max}} = 30, N = 14$, the throughput performance of ADMA in nanoNS3 is 90.2% on average, and 99.9% in best case, of that achieved using idealized simulator BSC; at $R_{\text{max}} = 70, N = 15$, throughput of ADMA is 89.1% on an

average and a maximum of 99% of that of BSC.

ADMA also acts as a multiple access control (MAC) mechanism. One of the requirements of a MAC protocol is fairness. Using simulations, we also calculate the fairness of *ADMA*. We calculate the Jain's fairness index [94] for different sets of $R_{\max}, N$ and p_t for both load aware and load unaware receivers. for $N = 14, R_{\max} = 30$ and $N = 15, R_{\max} = 70$ respectively. Fairness achieved by ADMA with deterministic receiver, on an average is 0.99 indicating high fairness.

Discussions and Future Work

The following assumptions were made in deriving the best address sequence for a network. We discuss in detail each of the assumptions below.

- p_t *is known*: As shown in Figure 4.9, the performance of an addressing sequence depends on the probability of sources transmitting bit 1. To select the best sequence, we must know the approximate range of p_t. Figure 4.11 plots the performance of an amplitude assignment algorithm when the receiver is unaware of input load; it assumes $p_t = 0.5$. The *integer sequence* under load unaware conditions performs close to that of a load aware deterministic receiver (within 95%). While the knowledge of p_t can improve the performance of the system, when using an *integer sequence* with a load unaware deterministic receiver, there is not a significant decrease in the throughput performance. Thus, p_t does not affect the performance of *ADMA*.

- p_t *is same across sources*: The insights developed in ADMA assume same p_t for all sources. Deriving the probability of success and the best sequence for different p_t across sources is a challenging problem. We showed that even when the receiver is not aware of p_t, throughput performance is not affected significantly. Following the same argument, if sources transmit bit 1 with a different probability, the performance of the system is not significantly affected.

- *Non empty data queue*: We implement OOK modulation, where an absence of a signal indicates bit 0. In practice, it is necessary to differentiate between bit 0 and *no data*. We propose the use of start and end-of-frame sequences. A pre-assigned bit sequence can define the start and end of a frame and an absence of signal outside this start and end of frames is considered as no-data.

- *Practical constraints on* R_{max}: The scalability of ADMA relies on the maximum amplitude and the number of amplitude levels that can be distinguished by the receiver. We beleive from the experiemental analysis that the receiver saturation is determined by the size of the receiver colony. Increasing the colony size can increase the number of receptors at the receiver allowing us to choose a higher R_{max}. The error in the response of the receiver bacteria for increasing amplitudes is not well understood and is a part of our future work.

CHAPTER 5

AWEC : AMPLITUDE-WIDTH ENCODING FOR ERROR CORRECTION IN BACTERIAL COMMUNICATION NETWORKS

It can be observed from the performance evaluation that ADMA is robust to amplitude errors that are much smaller than the amplitude difference between adjacent transmitters. As the number of transmitters increase, the bit error rate performance of ADMA deteriorates. An accurate detection of the transmitted amplitude is crucial in decoding both the address of the transmitter and the information transmitted by it. The focus of this work is to ensure reliable transfer of information in an MC system. Specifically, we focus on designing low complexity error correction mechanism that can be implemented using genetically engineered bacterial populations. The existing error correction codes are adapted from traditional coding techniques and do not consider the practicality of implementing these codes in a real-time, live bacterial system. The complexity of the code and its implementation is a crucial factor in the practical realization of the reliability mechanism. We analyze the different components of errors in an MC system and develop practical error correction mechanisms to correct each of these components.

Errors in an MC System

The amplitude of a molecular signal can be affected by the channel and the receiver design. We identify three types of errors that affect the the received amplitude viz., 1) channel induced errors, 2) receiver induced errors, and 3) collision-induced errors.

Channel error

Channel noise models [95, 16, 50] and capacity analysis [96, 97] have been developed for diffusive channel. Though the channel noise model depends on the geometry, type of molecule, rate of flow and varies with the application, we observe that in each of these

models, the channel noise is proportional to the concentration (amplitude) of the signal being transmitted. In this work, we consider a uniformly distributed channel noise model. The uniform distribution has maximum entropy in a bounded noise case and therefore has the worst case performance.

Receiver error

Successful decoding of bit 1 depends on the successful decoding of the amplitudes transmitted by each sender. As modeled and verified experimentally in [16], the response of a receiver bacteria to input rectangular signal is a non-linear function of the signal amplitude. The stochastic nature of receiver bacteria leads to variations in the fluorescent response of the receiver which in turn leads to receiver induced demodulation errors of the amplitude. The duration of the receiver response on the other hand, is a linear function of the duration of the input signal.

Collision error

When senders use OOK to transmit information, bit 0 is communicated by an absence of signal for a period T and therefore does not collide with signals from other senders at the receiver. Collision errors are caused by the transmission of bit 1 from multiple senders. The fewer the number of bit 1s to be transmitted, fewer is the number of collisions and hence few errors due to collisions. It must be observed that even though bit 0 from different sender does not collide, its reception can be affected by collision of bit 1.

Problem Definition and Design Challenges

To ensure reliable reception of the amplitude conveying both the information and the address, an error correction mechanism is required. Due to the high latency of an MC system [26], feedback based error correction mechanism will negatively impact the throughput performance and complexity of system design. Capacity approaching forward error correction (FEC) codes such as convolutional codes, LDPC [**11ac**] to detect and correct errors have been widely used in traditional networks. Implementing these FEC codes using bio-

logical circuits is highly challenging and the accuracy and consistency of the circuit design deteriorate with increasing complexity [57].

A number of research works have modified traditional FEC codes for MC networks without considering the practical constraints of an MC system [58, 59, 60]. [61] develops a family of ISI-free (Inter-Symbol Interference) codes that are simple and practical. The ISI free codes increase the Hamming distance between codewords and assign unique Hamming weight codewords to detect and correct codeword errors. Even though [61] provides a practical error correction code, it relies on a MAC protocol to handle collision. Any error caused by channel collisions will result in packet drop at the receiver.

An FEC that ensures reliable reception in a multiple access molecular communication network is still an open challenge. The focus of this work is in the design of a practical, low-complexity error correction code for an MC system with high latency.

Design of an efficient reliability mechanism has the following challenges

1. Low complexity : should not require additional modules to implement reliability mechanisms

2. High coding gain : should not affect the network throughput performance

3. High accuracy : correct amplitude errors induced by the channel and receiver

4. Multiple Access : should handle collisions in a multiple access network.

In this work, we develop a reliable molecular communication system that corrects for amplitude errors in a multiple access, single-hop network.

Amplitude-Width Forward Error Correction

Based on the insights from the analysis of amplitude errors, we develop AWEC, a forward error correction mechanism that embeds redundancy in the on-period and amplitude of the transmitted signal. The characteristics of a rectangular molecular signal as shown in

Equation 5.1 are, amplitude A, on-period T_{ON}, bit period T_b, and molecule type (AHL).

$$m(t) = \begin{cases} A, & 0 \leq t \leq T_{ON} \\ 0, & T_{ON} < t \leq T_b. \end{cases} \tag{5.1}$$

Receiver error correction

From the analysis of receiver induced error, we infer that the response of the receiver is a linear function of the duration of the signal. AWEC assigns a distinct on-period to different senders, thereby allowing accurate decoding of the sender at the receiver. Figure 5.1 is an illustration of distinct on-periods being assigned to two senders. On receiving the signal, the decoder maps the received T_{ON} to the closest on-periods that are assigned to the senders.

Algorithm 4 describes the AWEC encoder design. In lines 6-9, AWEC cyclically assigns the unique on-periods to each sender. The on-periods assigned are determined two system constraints specified by the receiver bacteria viz., T_{ON}, the minimum on-period required to decode a rectangular signal and w_e, an estimate of the width error at the receiver. The on-periods are chosen such that the minimum difference between two on-periods is $\geq \frac{w_e}{2}$ (line 8). A received on-period in the range $\left(T_1 - \frac{w_e}{2}, T_1 + \frac{w_e}{2}\right)$ is decoded as T_1 thus implying that the receiver can correct up-to $\frac{w_e}{2}$ error in the on-period of the transmitted sig-

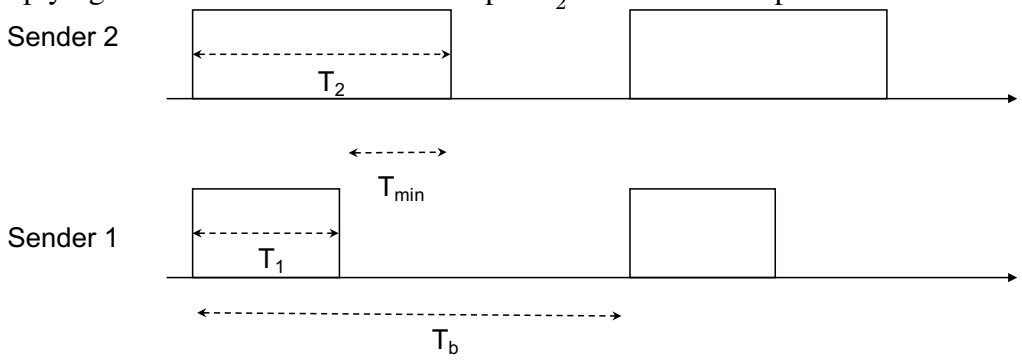


Figure 5.1: Redundancy in Duty-cycle

Algorithm 4 Encoder : Width and Amplitude Assignment

1: $N \leftarrow$ Number of senders
2: $N_w \leftarrow$ Number of unique on-periods
3: $(-\frac{w_e}{2}, +\frac{w_e}{2}) \leftarrow$ Estimated width error
4: $W \leftarrow \{T_1, T_2, \ldots, T_N\}$, on-period assigned
5: $A \leftarrow \{A_1, A_2, \ldots, A_N\}$, amplitudes
▷ Width assignment
6: **for** i := 1 to N **do**
7: bit_i $\leftarrow$ i($\mod N_w$)
8: $T_{\text{i}} \leftarrow T_{ON} +$ bit_i $\cdot\, w_e$
9: **end for**
▷ Amplitude assignment
10: **for** i := 1 to N **do**
11: block_i $\leftarrow$ i/N_w
12: $A_{\text{i}} \leftarrow 2^{\text{block\_i}}$
13: **end for**

nal. For a known bounded on-period error w_e, the receiver induced error can be corrected by increasing the distance between on-periods of senders.

However, assigning a unique on-period to each sender with a minimum distance of w_e increases the overall T_b, one bit-period. AWEC utilizes the amplitude of the rectangular signal to overcome this disadvantage of on-period based error correction. In line 7, AWEC limits the number of unique on-periods to N_w and assigns to N senders with repetition ($N_w < N$). The senders with the same on-period are differentiated using unique amplitudes i.e., AWEC assigns a unique 2-tuple id <**amplitude, on-period**> to each sender. The higher the number of on-periods, (N_w), lower is the intra-width collisions and higher the overall T_b. Here we present our solution to choose N_w that reduces the probability of intra-width collisions without reducing throughput significantly. The probability of k senders colliding at a given time is given by,

$$\Pr(k \textit{ collisions}) = \binom{N}{k} p^k \cdot (1-p)^{(N-k)} \tag{5.2}$$

where, N is the total number of senders in the network and p is the probability of each sender transmitting bit 1. The probability of intra-width collisions is calculated by replacing N with N_w in Equation 5.2. Choosing very high values of N_w affects the throughput

performance while very low values of N_w leads to increase in intra-width collisions.[1] We determine N_w that satisfies the condition of $\{\Pr(\textit{Intra-width collisions} \geq 2) \leq 0.2\}$ give p and N. The threshold can vary with application and the system in use. N_w that satisfies the above condition is then used in Algorithm 4 to assign on-periods.

Channel error correction

For a network with N sender and N_w unique on-periods, up-to $A_N = \lfloor \frac{N}{N_w} \rfloor$ senders have the same on-period (for each on-period). The choice of amplitudes assigned affect the error correction capability of AWEC due to channel and collision induced amplitude errors discussed in Section 5.2. It has been proved in [98] that a *binary set* of amplitudes, the set of increasing powers of 2 (example $S : \{1, 2, 4, 8\}$), is an optimal amplitude assignment to recover from collision errors. This is because the sum of any combination of powers of 2 is a unique value. For example, let four senders be assigned amplitudes $1, 2, 4, 8$ and an on-period T_1. On receiving an amplitude 5 and on-period T_1, the receiver can identify that senders with amplitudes 1 and 4 transmitted bit 1 while others transmitted bit 0, as there is only one possible way to arrive at this sum. Lines 10 to 13 in Algorithm 4 assigns powers of 2 amplitudes to the senders with the same on-period. Similar to on-period, the amplitudes are repeated across senders i.e., senders with different on-periods have same amplitude. Amplitude repetition is designed to allow for maximum distance between adjacent amplitudes.

A binary amplitude assignment can correct for collision errors but assumes that the amplitude received is accurate. In the presence of channel errors, the received amplitude can be greater than or less than the sum of amplitudes transmitted leading to an error in decoded amplitudes. The choice of N_w such that the probability of intra-width collisions is less than 0.2 implies that the probability of the received amplitude being the result of the sum of two amplitudes is very small. By choosing N_w that minimizes collisions, the absolute differ-

[1] To achieve fewer collisions and small values of N_w, the probability of each sender transmitting bit 1 must be small. In this work, we focus on a system with low probability of bit 1 (rectangular signal) transmitted by the sender. We consider encoding techniques [26] that achieve a low probability of bit 1 being transmitted in the channel such that p is low.

ence between amplitudes is used by the decoder to correct for channel-induced amplitude errors. In the example above, the probability of receiving an amplitude as the sum of 1 and 4 is very small. Therefore, the decoder finds the amplitude closest to 5 (4) and decodes bit 1 for sender with amplitude 4 and on-period T_1 and bit 0 for others. The exponentially increasing amplitudes is suitable to correct for amplitude dependent channel errors. In the binary set of amplitudes, as the amplitudes increase, the difference between adjacent amplitudes increases i.e., the minimum distance between adjacent amplitudes increases and therefore the error correction capability is unaffected. The maximum amplitude that can be assigned to a sender is limited by the saturation amplitude at the receiver. Let A_{max} be the maximum decodable amplitude i.e., any amplitude greater than A_{max} is received as A_{max}. Therefore, N_w must be chosen such that $2^{(N_w-1)} \leq A_{max}$.

Thus, AWEC performs error correction in two steps.

1. Inter-width errors : Receiver induced errors of upto $\frac{w_e}{2}$ that affects the on-period of the signal is corrected by increasing the distance between adjacent on-periods.

2. Intra-width errors : Channel induced amplitude errors of upto $\frac{A_{min}}{2}$ that affects the amplitude of the received signals is corrected by increasing the minimum distance between amplitudes that share the same on-period.

AWEC Codewords

To this end, we presented AWEC embedding redundant information in the on-period and amplitude of the signal to achieve reliability. Embedding redundancy in the transmitted signal offers a practical implementation of codeword generation in an MC system. The discrete samples of the transmitted signal with a given $< \mathbf{amplitude}, \mathbf{on-period} >$ is the codeword while the samples with $< 1, T_{ON} >$ is the actual signal. By increasing the on-period and the amplitude values, AWEC can generate codewords without any need for complex mathematical operation. Senders with different on-periods thus transmit rectangular signals whose samples are "1" for the assigned T_{ON}. For example, consider $T_b = 5$

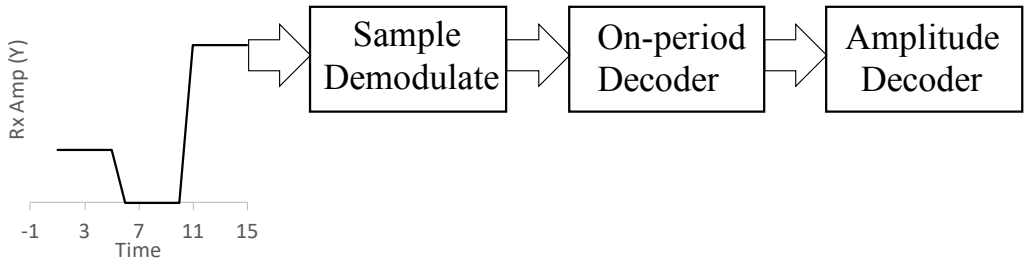


Figure 5.2: Decoder Architecture

min, a sample period of 1 minute and $T_1 = 3$ min and $T_2 = 5$ min represent codewords $\{1, 1, 1, 0, 0\}$ and $\{1, 1, 1, 1, 1\}$ of bit 1 for sender 1 and sender 2 respectively; $\{0, 0, 0, 0, 0\}$ represent bit 0 for every sender. The hamming distance between two codewords is the number of samples to represent w_e (here, $w_2 = 2$). Similarly, two senders with same T_{ON} are assigned unique amplitudes. For example, assigning amplitudes 2 and 4 to senders with $T_{ON} = 3$ implies their respective bit 1 codewords are $\{2, 2, 2, 0, 0\}$ and $\{4, 4, 4, 0, 0\}$. The absolute difference between the amplitudes determines intra-width error correction capability of AWEC.

Amplitude-Width Decoder

The receiver samples, demodulates and decodes the receiving samples to identify the amplitude and on-period of the signal. In this section, we present the decoder architecture used in AWEC to correct for errors with apriori knowledge of the encoder and the system.

Sample and Demodulate:

The first block of the decoder architecture shown in Figure 5.2 is *sample and demodulate*. We utilize the inverse of the receiver response model proposed in [16] to perform sampling and demodulation. The model proposed in [16] is experimentally validated and output the response of a receiver bacteria to an input chemical signal. The inverse of the model takes as input the response of the bacteria and output the amplitude samples for a given sampling rate. The output of the inverse model is thus a time sequence of amplitude samples at the receiver. Each sample in this sequence is the sum of amplitudes from differ-

ent senders at a particular time instant and the amplitude error introduced by the receiver and channel.

An AWEC decoder takes these samples and estimates the codewords that were transmitted by each sender and corrects for channel and receiver errors. These codewords are then used by the MAC decoder to detect and correct collision errors. The output of *sample and demodulate* module is the time sequence of samples. If T_{ON} is the minimum time difference between on-periods and to capture the on-periods, the sampling rate must be at least $\frac{1}{2T_{ON}}$. The sample and demodulate module uses this parameter to generate $\frac{1}{2T_{ON}}$ discrete samples per second.

On-period Decoder: The next step in the decoder architecture is to identify the on-period and correct for receiver errors. The *on-period decoder* deciphers the reception of a signal from the amplitude transitions between adjacent samples. Let $s_0, s_1, s_2, \ldots$ be the time sequence of received samples and $r_1, r_2, \ldots$ be the differences of the received sample sequence i.e., $r_1 = s_1 - s_0$. A positive value of r_i indicates the reception of bit 1 by one or more senders and the corresponding difference r_i is the received amplitude. The *on-period decoder* searches for a matching $-r_i$ within one bit period. The difference in time at which the positive and negative r_is were received is the received on-period. Identifying the corresponding negative transition is simpler when the senders are time synchronized and transmit bits only in predetermined slots. The challenge in an unsynchronized system is in identifying the rise and fall of the rectangular signals when multiple senders are transmitting simultaneously. We present a decoder that considers each case of sender collisions and corrects for collision,, receiver and channel induced errors. We identify the scenarios where the decoder cannot detect or correct for errors. The bit period T_b remains constant across senders, and therefore, r_i, the rise in amplitude corresponding to bit 1 transmission from one(or more) sender must have a corresponding $-r_i$ that indicates the fall of the signal within T_b from r_i. Thus, for every positive rise in the sample amplitude difference, the decoder searches for a matching negative fall with the amplitude. If the rise and fall am-

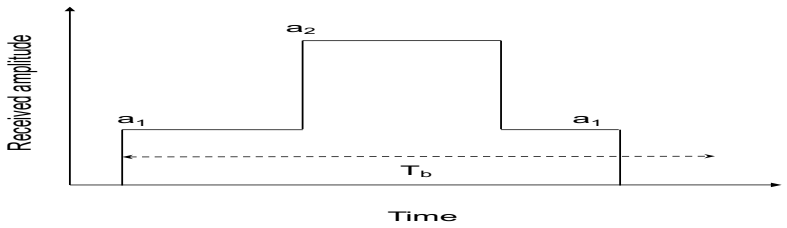


Figure 5.3: Received Sample Sequence : Illustration

plitudes match, the decoder finds the difference in location/time of the matching rise and fall samples to estimate the received on-period $\hat{T_{ON}}$. Figure 5.3 is an illustration of the rise and fall transition. The two positive transitions a_1 and a_2 are matched with the negative transitions $-a_1$ and $-a_2$ respectively in Figure 5.3.

When rise and fall of a signal do not collide with the rise and/or of another signal, the decoder can uniquely identify the on-periods and their corresponding amplitude transitions r_i by parsing through the received samples sequentially (Figure 5.3). In this case, the rise and fall are uniquely identifiable even if the samples in between collide with other signals. However, as the number of senders increases, the probability of two senders colliding with the rise and/or fall increases. We present a heuristic decoder algorithm that iterates through each case and estimates the rise and fall of a received signal. For every positive difference $r_i > 0$, the decoder considers the next $2T_b$ samples(the samples corresponding to T_b) and checks for each of the following scenarios to search for the corresponding fall and decode the on-period. When one of the cases returns true, the on-period decoder exits the search and proceeds to decode the amplitude and then to the next rise transition. Algorithm 5 is the pseudo-code for the decoder implementation. The receiver parses **dataRx**, the output of sample and demodulate and identifies amplitude transitions (line 41 to 43). For each positive transition or rise, a matching fall is identified and the on-period and amplitude are decoded. The function **FindMatchingFall()** searches for a fall within *stopPos*, T_b period from *startPos*, start of the transition. The decoder first searches for a fall that matches the rise amplitude (lines 17 to 22). If more than one fall occurs within T_b, the decoder check each of the fall to find a valid on-period (line 18). The function **OnPeriodDecoder** checks

if the estimated on-period is valid by verifying the difference between estimated on-period $\hat{T_{ON}}$ and on-periods assigned to senders (line 1 to 5). The fall whose corresponding on-period is valid is decoded as the matching fall and passed on to the *AmplitudeDecoder*. The amplitude decoder is similar to an on-period decoder. It compares the amplitude difference between the current sample and previous sample (*r*xAmp) with the amplitudes assigned to sender with the decoded on-period. If the decoded amplitude is valid, the decoder updates the amplitude and on-period of the received samples for the corresponding sender.

When no such matching fall is identified, the decoder searches for rise collisions i.e., the decoder checks if more than one signal collides and starts at the same time while still having different fall positions(lines 23 to 29). In this case, no single fall will match the rise amplitude, but, a combination of falls that correspond to the colliding rise signals will match with the rise observed. To identify this case, the decoder generates a array *sumsOfFall* of the sum of subsets of falls in the range dataRx[startPos:stopPos] considered. If signals collided only at rise, and have a distinct falls, the rise will find a matching entry in *sumsOfFall* (line 23). The decoder loops through each fall value corresponding to their summation (*fallCombinations*) and estimates the on-period, verifies it validity and decodes the amplitude.

If the decoder cannot find any matching entry for the rise in sumsOfFall, it repeats the steps to identify for fall collisions (lines 30 to 36). If more than one sender ends or stops transmitting their signal at the same time, while still beginning at different times, then the above cases return false. A combination of rise amplitudes then match with a single fall. The decoder follows the same steps as that of fall collision check to find rise collision check by replacing sumsOfFall by sumsOfRise. The decoder returns if all three cases fail, the decider returns without being able to detect or correct any error and moves to the next step.

All collisions except the following can be corrected by the above AWEC decoder architecture.

1. Two senders with the same amplitude transmit bit 1 one after the other such that the fall of sender 1 collides with the rise of sender 2. If the sum of on-periods is closer to another on-period, such a collision will be decoded incorrectly.

2. Two senders with same amplitude rise and fall within T_b. If $a_1 = a_2$ in Figure 5.3, the decoder cannot match the rise and fall using amplitudes.

Amplitude Decoding and Error Correction: The output of the on-period decoder provides the estimate of the decoded on-period $\hat{T_{ON}}$, the time of rise and fall of the received signal and the corresponding r_i. The *amplitude decoder* resolves the transmitted amplitudes with prior knowledge of the amplitudes assigned to senders, estimated channel error and the received amplitude r_i. For each on-period, the amplitude decoder stores the list of amplitude assigned. The amplitude decoder finds the amplitude assigned (example $\{1, 2, 4, 8\}$) that is closest to the received amplitude. On receiving an amplitude 5, the amplitude decoder output is set to 4 for the positions corresponding to rise and fall time that is output by the on-period decoder. The output of the amplitude decoder is thus the decoded samples for each sender.

We make use of randomness in the asynchronous transmissions in decoding signals from multiple senders. Since the senders are not time synchronized, the probability of k collisions derived in Equation 5.2 is further reduced by the random delay in the start of a message. Inter-period collision errors are corrected by the redundancy introduced in duty-cycle and the duty-cycle decoder design. Amplitude decoding is performed assuming that the received amplitude (for a decoded on-period) is from a single sender and therefore the amplitude decoder corrects for channel errors.

To this end, we have discussed the system constraints and challenges in implementing an error correction mechanism in an MC system. We have presented AWEC, a practical and easy to implement error correction mechanism that embeds redundancy in the characteristics of the signal.

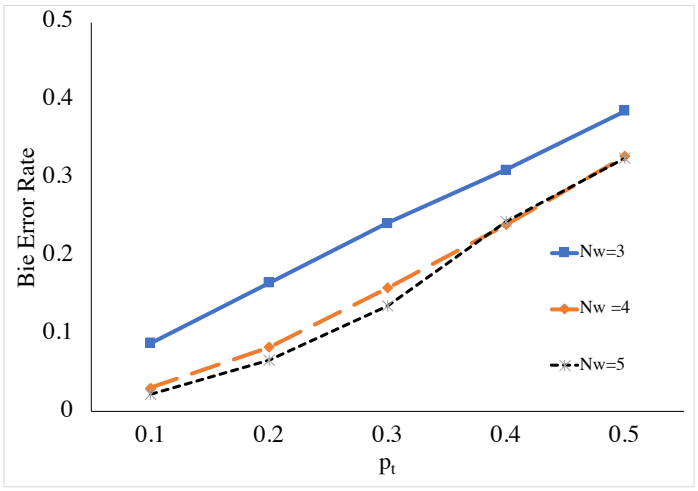


Figure 5.4: Bit Error Rate Performance of AWEC, $N = 15$

AWEC Performance Evaluation

We built a python based Bacterial Communication Simulator (BCS) to evaluate the performance of the error correction techniques presented here. We implement OOK as modulation technique where every sender transmits bit 1 with a probability p_t. BCS implements the encoder and decoder presented in Sections 5.3 and 5.4 respectively. We simulate the response of the receiver bacteria using the model developed in [16]. The inverse of the model is used to perform sampling and demodulation.

Unless otherwise mentioned, we use the following parameters in the simulations. A uniformly distributed, bounded amplitude error that is proportional to the amplitude of the transmitted signal is added randomly to the transmitted signal i.e., an amplitude a_i after passing through the channel and receiver, is received as $a_i + \epsilon a_i$, where ϵ is the percentage of error introduced by the channel. We assume that the error percentage is the same across senders without loss of generalization. AWEC implementation does not change for varying amplitude errors. We consider a 20% channel error in our evaluations. A_{max} is the receiver saturation amplitude. Any amplitude $\geq A_{max}$ is received as A_{max} by the receiver bacteria.

Figure 5.4 shows the average bit error rate performance of the network AWEC as a function of source load distribution p_t, for $A_{max} = 45, N = 15$. The three curves represent the performance for increasing values of N_w. For the given setting, the minimum value of N_w

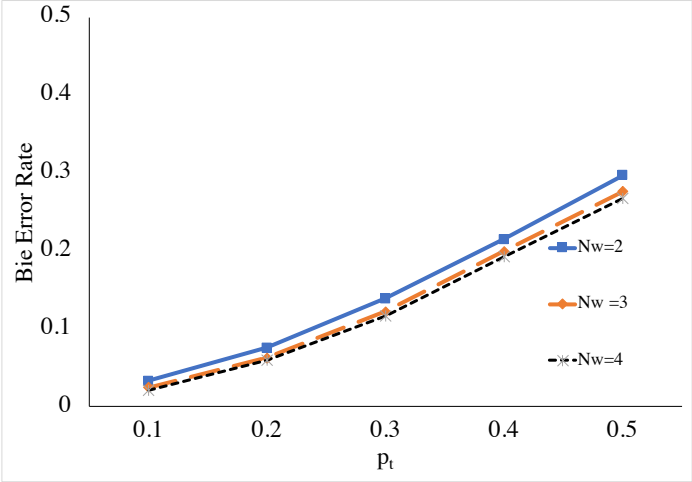


Figure 5.5: Bit Error Rate Performance of AWEC, $N = 10$

is 3. As we increase N_w, the bit error rate for a given p_t is decreases. This is explained by the reduction in intra-width collisions with increasing N_w. The higher the N_w, lesser is the number of senders with the same on-period. However, as N_w increases, the average throughput performance of a single sender compared to that of the maximum throughput using OOK decreases. As N_w increases, the bit-period to accommodate all on-periods increases, thus decreasing the overall throughput. Though by allowing multiple senders to transmit simultaneously, the network throughput is improved, individual throughput performance is traded off to improve bit error performance. We repeat the exercise for $N = 10$ in Figure 5.5. For the reduced number of senders, the minimum value of N_w is also reduced. For a fewer number of senders, the probability of collision is smaller and hence the overall bit error rate is smaller than $N = 15$.

In both these cases, the error correction capability of AWEC decreases with increasing value of p_t i.e., the bit error rate increases with increasing p_t. This is attributed to collision errors that dominate overall errors at higher values of p_t. As discussed in Section 5.3, in a high latency, low complexity MC system, it is desirable to use codewords with low weight i.e., to use application, message encoding and modulation techniques such that the probability of bit 1 in the channel is very small. The higher the probability of collisions, higher is the value of N_w required to achieve a lower probability of intra-width collision. At $p_t = 0.1$, the bit error rate of AWEC is of the order of 10^{-2} for $N = 15, A_{max} =$

$45, N_w = 3$ and 10^{-3} for $N = 10, A_{max} = 45, N_w = 2$. AWEC improves the bit error rate performance by an order of magnitude from simple ADMA by an order of magnitude. The improvement in the error correction capability is achieved at the cost of reduced throughput due to increased bit-period.

Algorithm 5 Decoder : On-period and Amplitude Estimation

function ONPERIODDECODER(risePos,fallPos)
    $\hat{T_{ON}} \leftarrow \mathbf{index}(\text{fallPos}) - \mathbf{index}(\text{risePos})$
    closestWidth $\leftarrow \mathbf{min}(\text{width} - \hat{T_{ON}})$
    $\mathbf{isValid}(\text{closestWidth})$
end function
function AMPLITUDEDECODER
    $\hat{a} \leftarrow rxAmp$, closestAmp $\leftarrow \mathbf{min}(\text{Amplitude} - \hat{a})$
    $\mathbf{isValid}(\text{closestAmp})$
end function
function FINDMATCHINGFALL(startPos,stopPos))
    diffRx $\leftarrow \mathbf{diff}(\text{dataRx}[\text{startPos:stopPos}])$
    allRise $\leftarrow$ diffRx$[\mathbf{where}(diffRx > 0)]$
    allFall $\leftarrow$ diffRx$[\mathbf{where}(diffRx < 0)]$
    firstRise $\leftarrow$ allRise[0], firstFall $\leftarrow$ allFall[0]
    ▷ A distinct fall is found with T_b from firstRise
    if firstRise **in** allFall **then**
        for fallPos **in index**(allFall == firstRise) **do**
            $\mathbf{OnPeriodDecoder}(\mathbf{index}(\text{firstRise}), \text{fallPos})$
            $\mathbf{AmplitudeDecode}()$
        end for
        return
    ▷ Check for collisions at the rise of signals
    else if firstRise **in** sumsOfFall **then**
        fallCombinations $\leftarrow \mathbf{elements}(\text{sumsOfFall})$
        for fall **in** fallCombinations **do**
            $\mathbf{OnPeriodDecoder}(\mathbf{index}(\text{firstRise}), \text{fall})$
            $\mathbf{AmplitudeDecode}()$
        end for
        return
    ▷ Check for collisions at the fall of signals
    else if firstFall **in** sumsOfRise **then**
        riseCombinations $\leftarrow \mathbf{elements}(\text{sumsOfRise})$
        for rise **in** riseCombinations **do**
            $\mathbf{OnPeriodDecoder}(\mathbf{index}(\text{rise}), \text{firstFall})$
            $\mathbf{AmplitudeDecode}()$
        end for
        return
    end if
end function
    ▷ Main Function
for sample **in** dataRx **do**
    prevSample $\leftarrow$ dataRx$[\mathbf{index}(\text{sample}) - 1]$
    if sample $-$ prevSample > 0 **then**
        startPos $\leftarrow \mathbf{index}(\text{sample})$
        stopPos $\leftarrow$ startPos $+ T\_b$
        FindMatchingFall(startPos,stopPos,sample)
    end if
end for

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

In this dissertation, we considered a bionetwork with genetically engineered bacteria as the sensor, the transmitter, and the receiver, and developed algorithms to build a single-hop bionetwork. There are several open challenges and problems that must be addressed to realize our vision of an autonomous network of bio-sensors. The communication algorithms we have developed are a small, but significant effort toward building a network of biosensors. Bionetworks can potentially reach into domains that would otherwise remain partially or completely inaccessible due to various challenges such as compatibility, safety, and lack of infrastructure, among others. For example, real-time monitoring of the functioning of the human body with minimally invasive techniques, learning and understanding the interaction between disease-causing pathogens and their surroundings, autonomous targeted drug delivery, and therapeutics. Recent advancements in synthetic biology have led to the design of logic gates, storage devices and computing units using live biological entities such as bacteria [99, 14, 70, 100]. However, in its current design, biological computing devices have very high processing delays, and their computational capacity is much lower than that of electronic processors. Therefore, existing communication algorithms cannot be applied directly to bionetworks. Our current work on communication algorithms leads to the following research challenges, which in turn broadens the scope of applications of molecular communication.

Modulation:

TEC, the modulation technique we developed, is suitable for biolinks with very high processing delays. When the channel conditions vary and have timing errors comparable to the processing delays, TEC is unreliable and TEC-SMART will be inefficient. Novel and practical modulation techniques that incorporate the constraints of the biolink based on the

application are required. A detailed understanding and the analysis of the channel conditions and system constraints are essential to the design of efficient modulation techniques.

Channel Coding:

TEC-SMART performs error correction assuming the nodes are static. In practice, we expect biosensors to be dynamic and mobile, varying the length of the biolink, which in turn alters the estimated time elapsed between signals. Algorithms for error detection and correction mechanisms for dynamic and mobile nodes are open challenges. Depending on the application, the acceptable bit error probabilities can vary, which can be traded off for increased throughput. Research on channel codes that are robust to the variations in the biolink is required to build an autonomous biosensor network.

Data Compression:

In this dissertation, we developed communication algorithms with a philosophy of *talk less say more* due to high delays in a bacterial communication system. In systems with extremely high latency, transmitting a signal in the channel is expensive in terms of delay and throughput. The maximum amount of information must be transferred with a fewer number of signals. This work opens future research on efficient and practical source coding techniques that can be implemented using biological systems to perform data compression and maximize information transfer. Computationally intensive compression algorithms have been developed for traditional networks to reduce redundancy in information transmitted. Innovative compression algorithms for devices with low computational complexity remain an open challenge for future work.

System Design:

The basic components of a communication system are the encoder, the modulator, the demodulator, and the decoder. The encoder (and the decoder) further includes a source encoder and a channel encoder to perform compression and error correction respectively. We developed communication algorithms with proof-of-concept design for some of the components. For example, we used an accurate channel and receiver response model to

perform sampling and demodulation; however, it is not practical to develop an accurate model of each receiver. Practical realization of each component is essential to build a biolink. System design will rely on successful collaborations with teams from synthetic biology, bioengineering, and mechanical engineering.

MAC:

ADMA implicitly solves MAC in a star topology. Algorithms to solve MAC in a topology with more than one receiver is an open challenge. Implementing random access methods such as listen-before-talk, can increase the latency significantly; also, the design of transceivers to switch from receive to transmit mode is challenging. MAC protocols that incorporate the constraints of different bionetworks and leverage the new opportunities presented by the biolink will be a focus of my future research.

Routing:

As the network grows, and communication to a receiver is through multiple hops, routing algorithms will be required to find the best route to the destination. Based on the application and network topology, constraints in the design of a routing algorithm can change. High latency and low complexity make it challenging to design a feedback link, making network discovery difficult. Due to the dynamic nature of the transceivers and the channel, routing algorithms must be capable of adapting to node failures and route changes.

Reliability:

AWEC embeds redundancy in the on-period of the transmitted signal. By increasing the number of sensors, the overall bit period increases, which then decreases the network throughput. AWEC is therefore not suitable for networks with a large number of sensors as it reduces overall network throughput. Capacity-approaching channel codes have been developed for traditional networks. These codes are not practical to implement in a bionetwork. An error correction code that approaches maximum channel capacity and is practical to implement in bionetworks remains an open challenge. As the channel model varies with the application and the system setup, error correction codes that can adapt to

different channel conditions are needed.

The above challenges are specifically applicable to a category of bionetworks called bio-only network where the sensor, the transmitter, and the receiver are biological entities. Working towards building a bionetwork, this research also opens two broad categories of bionetworks viz., 1) Bio-Electronic Networks 2) Bio-Electronic Systems, based on the sensor, the transmitter, and the receiver used. Each of these classes has applications on their own while also advancing us towards bio-only networks. The constraints and opportunities of each of these categories pose new challenges in the design of communication algorithms and practical implementation.

Bio-Electronic Networks

A bio-electronic network consists of a biosensor, an electronic transmitter, and an electronic receiver. Examples include wearable sensors with a wireless network card [8], fitness trackers, and vital sign monitors [5]. Research programs on environmental sensing, wearable sensors, and healthcare connectivity funded by agencies such as the National Science Foundation (NSF) and the National Institutes of Health (NIH) are indicative of the growing need and interest in the research community towards bio-electronic networks [101, 102, 6, 103]. Currently, bio-electronic networks utilize the existing wireless (Bluetooth, WiFi) infrastructure for data transmission. However, data transmission in applications involving biosensors presents unique challenges and opportunities for the design of practical and efficient communication algorithms. Such algorithms developed for resource-constrained nodes are also applicable to other domains such as IoT, ad-hoc networks, and vehicular networks.

Asymmetric Communication: In existing communication systems, signal processing, storage, and data transmission are all handled at the transmitter end. This creates an asymmetry in the work done in the system; transmitter being at the heavy end and the receiver at the light end. In a bio-electronic network, the transmitter is attached to a biosensor and

has limited battery power, computational resources, and storage, making it the weaker end. Such a workload and resource asymmetry with the weaker node carrying out the heavy lifting is inefficient. Limited memory at the transmitter also leads to a knowledge asymmetry. For example, in an environmental monitoring system, the receiver collects information from multiple transmitters and it also has access to past data, which allows it to predict future measurements. Communication algorithms designed to address and leverage such asymmetries will improve the durability and performance of the transmitter as well as the overall system.

Coexistence and Cooperation A variety of bio-electronic systems have been used in health care [7] and precision farming [104] for monitoring and diagnostics. Typically, in such kind of monitoring applications, the number of sensors deployed is large while the amount of information transmitted by each of them is small. In such high-density environments with resource-constrained transmitters, solutions to address radio resource management and medium access control are required. Scheduling-based algorithms to achieve energy-efficient, spectrum-efficient and scalable coexistence of biosensors [105] can lead to efficient coexistence. In a typical bio-electronic network, information from multiple sensors can be highly correlated as the sensors are deployed in close proximity to each other. Practical, scalable, and resource-efficient cooperative communication algorithms that can reduce such kind of information redundancy is required.

Bio-Electronic Systems

A bio-electronic system consists of a biosensor, a biotransmitter, and an I/O device for electronic read-out. Examples include a wearable sensor with genetically engineered bacteria [106] that emit fluorescence in response to chemical stimuli from the human body and blood tests using lab-on-chip [7] that makes use of biochemical reactions to monitor for nutrient deficiency. The output, which depends upon the sensor design, is captured by a suitable device such as a phone's camera or a microscope and processed to extract infor-

mation. The efficiency of a bio-electronic system thus depends not only on the biosensor, but also on the read-out mechanism, the architecture, shape, and location of the sensor.

This system demands a redesign of sensor architecture and response mechanism of biotransmitter in order to obtain information more quickly, accurately and efficiently. Though the design of a biosensor depends strongly on the specific application, the response (reporting) methods of the transmitter can be designed to improve the overall efficiency of communication. A number of ongoing research programs funded by the NSF and the NIH reveal a promising trend towards the development of programmable biomarkers and read-out mechanisms [107, 108, 109]. My work on TEC [26], which proposes a response technique that minimizes the number of signals broadcast in order to convey a message, can be applied in such scenarios. The response of the biotransmitter to a sensor signal conveys information to the receiver. Biotransmitter design is analogous to modulation technique in a traditional communication system. Alternate response mechanisms for biotransmitters such as the color of fluorescence (convey different messages as a function of the fluorescence response), varying the densities of molecules in response to a stimulus, which can then tracked using post-processing techniques such as image processing can open up more applications. In applications such as healthcare and food and water quality monitoring, the accuracy of the information sensed and reported is crucial. Designing biosensors that can accurately detect a stimulus is the first step towards achieving this. In a bio-electronic system, the response of the biotransmitter to the output of the biosensor also affects the overall accuracy of the system accuracy, and this response can be (indirectly) affected by cross-talk between the transmitters as well as interference from other stimuli in the surroundings.

Information fidelity can be improved by introducing redundancy in the transmitter response with the help of multiple biotransmitters and a spatial arrangement of sensors that can minimize cross-talk and noise from the surroundings. While, in principle, reliability algorithms in traditional networks have the same goal of improving signal fidelity, existing reliability solutions rely on redundancy in the bits transmitted. Algorithms to improve

signal fidelity in bio-electronic systems by embedding reliability mechanisms in the design and architecture of the sensor and the transmitter, thus reducing the computational burden on the biological circuits of the transmitter is required. In practice, multiple biotransmitters can be used to broadcast multiple signals at the same time. Read-out mechanisms and transmitter architecture that can process this parallel information broadcast are needed. In ADMA, a unique amplitude was assigned to each transmitter and used by the receiver to identify the transmitters. I plan to focus on alternate, novel read-out mechanisms that will allow for faster and parallel reporting. Bio-electronic systems will further the scope for collaboration with teams from information security (to securely broadcast data) and data science (to process and understand the received data).

REFERENCES

[1] A. Jain, A. Graveline, A. Waterhouse, A. Vernet, R. Flaumenhaft, and D. E. Ingber, "A shear gradient-activated microfluidic device for automated monitoring of whole blood haemostasis and platelet function," *Nature communications*, vol. 7, 2016.

[2] M. Di Lorenzo, A. R. Thomson, K. Schneider, P. J. Cameron, and I. Ieropoulos, "A small-scale air-cathode microbial fuel cell for on-line monitoring of water quality," *Biosensors and Bioelectronics*, 2014.

[3] W. Zhang, A. M. Asiri, D. Liu, D. Du, and Y. Lin, "Nanomaterial-based biosensors for environmental and biological monitoring of organophosphorus pesticides and nerve agents," *TrAC Trends in Analytical Chemistry*, 2014.

[4] *Darpa grant:* `https://www.fbo.gov/index?s=opportunity&mode=form&id=620a8c69a46dc688d98aa6c1af08449d&tab=core&_cview=0`.

[5] *Eccrine systems:* `https://www.eccrinesystems.com/`.

[6] *Nsf grant:* `https://www.nsf.gov/awardsearch/showAward?AWD_ID=1642513&HistoricalAwards=false`.

[7] *Nutriphone*, `http://insight.cornell.edu/nutriphone`.

[8] Y. Khan, M. Garg, Q. Gui, M. Schadt, A. Gaikwad, D. Han, N. A. Yamamoto, P. Hart, R. Welte, W. Wilson, *et al.*, "Flexible hybrid electronics: Direct interfacing of soft and hard electronics for wearable health monitoring," *Advanced Functional Materials*, 2016.

[9] "Electromagnetic wireless nanosensor networks," *Nano Communication Networks*,

[10] G Deligeorgis, M Dragoman, D Neculoiu, D Dragoman, G Konstantinidis, A Cismaru, and R Plana, "Microwave propagation in graphene," *Applied Physics Letters*,

[11] T. Suda, M. Moore, T. Nakano, R. Egashira, A. Enomoto, S. Hiyama, and Y. Moritani, "Exploratory research on molecular communication between nanomachines," in *GECCO 2005*.

[12] I. F. Akyildiz, F. Fekri, R. Sivakumar, C. R. Forest, and B. K. Hammer, "Monaco: Fundamentals of molecular nano-communication networks," *IEEE Wireless Communications*,

[13] N. Farsad, A. Eckford, S. Hiyama, and Y. Moritani, "On-chip molecular communication: Analysis and design," *NanoBioscience, IEEE Transactions on*, 2012.

[14] T. Danino, Mondragon-Palomino, Octavio, Tsimring, *et al.*, "A synchronized quorum of genetic clocks.," *Nature*, 2010.

[15] D. M. Shcherbakova, A. A. Shemetov, A. A. Kaberniuk, and V. V. Verkhusha, "Natural photoreceptors as a source of fluorescent proteins, biosensors, and optogenetic tools," *Annual review of biochemistry*, 2015.

[16] C. M. Austin, W. Stoy, P. Su, M. C. Harber, J. P. Bardill, B. K. Hammer, and C. R. Forest, "Modeling and validation of autoinducer-mediated bacterial gene expression in microfluidic environments," *Biomicrofluidics*, 2014.

[17] A. G. Brolo, "Plasmonics for future biosensors," *Nature Photonics*, 2012.

[18] J. Tamayo, P. M. Kosaka, J. J. Ruz, Á. San Paulo, and M. Calleja, "Biosensors based on nanomechanical systems," *Chemical Society Reviews*, 2013.

[19] Q. Liu, C. Wu, H. Cai, N. Hu, J. Zhou, and P. Wang, "Cell-based biosensors and their application in biomedicine," *Chemical reviews*,

[20] D. Endy, "Foundations for engineering biology," *Nature*,

[21] B. L. Bassler, "How bacteria talk to each other: Regulation of gene expression by quorum sensing," *Current Opinion in Microbiology*,

[22] P. Melke, P. Sahlin, A. Levchenko, and H. Jönsson, "A cell-based model for quorum sensing in heterogeneous bacterial colonies," *PLoS Computational Biology*, 2010.

[23] *Sample6:* `https://www.sample6.com/`.

[24] J. Nemunaitis, C. Cunningham, N. Senzer, J. Kuhn, J. Cramm, C. Litz, R. Cavagnolo, A. Cahill, C. Clairmont, and M. Sznol, "Pilot trial of genetically modified, attenuated salmonella expressing the e. coli cytosine deaminase gene in refractory cancer patients," *Cancer gene therapy*, vol. 10, no. 10, p. 737, 2003.

[25] A. Schroeder, D. A. Heller, M. M. Winslow, J. E. Dahlman, G. W. Pratt, R. Langer, T. Jacks, and D. G. Anderson, "Treating metastatic cancer with nanotechnology," *Nature Reviews Cancer*, vol. 12, no. 1, p. 39, 2012.

[26] B. Krishnaswamy, C. M. Austin, J. P. Bardill, D. Russakow, G. L. Holst, B. K. Hammer, C. R. Forest, and R. Sivakumar, "Time-elapse communication: Bacterial communication on a microfluidic chip," *Communications, IEEE Transactions on*,

[27] T. Danino, A. Prindle, G. A. Kwong, M. Skalak, H. Li, K. Allen, J. Hasty, and S. N. Bhatia, "Programmable probiotics for detection of cancer in urine," *Science translational medicine*, vol. 7, no. 289, 289ra84–289ra84, 2015.

[28] A. Tamsir, J. J. Tabor, and C. A. Voigt, "Robust multicellular computing using genetically encoded nor gates and chemical wires," *Nature*, vol. 469, no. 7329, p. 212, 2011.

[29] B. Wang, R. I. Kitney, N. Joly, and M. Buck, "Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology," *Nature communications*, vol. 2, p. 508, 2011.

[30] P. Siuti, J. Yazbek, and T. K. Lu, "Synthetic circuits integrating logic and memory in living cells," *Nature biotechnology*, vol. 31, no. 5, p. 448, 2013.

[31] J. Q. Boedicker, M. E. Vincent, and R. F. Ismagilov, "Microfluidic confinement of single cells of bacteria in small volumes initiates high-density behavior of quorum sensing and growth and reveals its variability," *Angewandte Chemie International Edition*, vol. 48, no. 32, pp. 5908–5911, 2009.

[32] P. Guo, E. W. Hall, R. Schirhagl, H. Mukaibo, C. R. Martin, and R. N. Zare, "Microfluidic capture and release of bacteria in a conical nanopore array," *Lab on a Chip*, vol. 12, no. 3, pp. 558–561, 2012.

[33] A. Y. Yong, S. Shabahang, T. M. Timiryasova, Q. Zhang, R. Beltz, I. Gentschev, W. Goebel, and A. A. Szalay, "Visualization of tumors and metastases in live animals with bacteria and vaccinia virus encoding light-emitting proteins," *Nature biotechnology*, vol. 22, no. 3, p. 313, 2004.

[34] L. You, R. S. Cox III, R. Weiss, and F. H. Arnold, "Programmed population control by cell–cell communication and regulated killing," *Nature*, vol. 428, no. 6985, p. 868, 2004.

[35] S. MacGregor, N. Rowan, L McIlvaney, J. Anderson, R. Fouracre, and O Farish, "Light inactivation of food-related pathogenic bacteria using a pulsed power source," *Letters in Applied Microbiology*, vol. 27, no. 2, pp. 67–70, 1998.

[36] D. M. Wolf, L. Fontaine-Bodin, I. Bischofs, G. Price, J. Keasling, and A. P. Arkin, "Memory in microbes: Quantifying history-dependent behavior in a bacterium," *PLOS one*, vol. 3, no. 2, e1700, 2008.

[37] F. Farzadfard and T. K. Lu, "Genomically encoded analog memory with precise in vivo dna writing in living cell populations," *Science*, vol. 346, no. 6211, p. 1 256 272, 2014.

[38] S. L. Shipman, J. Nivala, J. D. Macklis, and G. M. Church, "Molecular recordings by directed crispr spacer acquisition," *Science*, vol. 353, no. 6298, aaf1175, 2016.

[39] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, p. 335, 2000.

[40] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, no. 6767, p. 339, 2000.

[41] A. E. Friedland, T. K. Lu, X. Wang, D. Shi, G. Church, and J. J. Collins, "Synthetic gene networks that count," *Science*, vol. 324, no. 5931, pp. 1199–1202, 2009.

[42] M. Pierobon and I. F. Akyildiz, "A physical end-to-end model for molecular communication in nanonetworks," *Selected Areas in Communications, IEEE Journal on*,

[43] B. Atakan and O. B. Akan, "On channel capacity and error compensation in molecular communication," in *Transactions on computational systems biology X*, Springer.

[44] A. Einolghozati, M. Sardari, and F. Fekri, "Design and analysis of wireless communication systems using diffusion-based molecular communication among bacteria," *Wireless Communications, IEEE Transactions on*, 2013.

[45] M. Pierobon and I Akyildiz, "A statistical-physical model of interference in diffusion-based molecular nanonetworks," 2014.

[46] N.-R. Kim and C.-B. Chae, "Novel modulation techniques using isomers as messenger molecules for nano communication networks via diffusion," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 847–856, 2013.

[47] M. S. Kuran, H. B. Yilmaz, T. Tugcu, and I. F. Akyildiz, "Modulation techniques for communication via diffusion in nanonetworks," in *2011 IEEE international conference on communications (ICC)*, IEEE, 2011, pp. 1–5.

[48] M. U. Mahfuz, D. Makrakis, and H. Mouftah, "Spatiotemporal distribution and modulation schemes for concentration-encoded medium-to-long range molecular communication," in *Communications (QBSC), 2010 25th Biennial Symposium on*, IEEE.

[49] C. Rose and I. S. Mian, "A fundamental framework for molecular communication channels: Timing & payload," in *IEEE ICC*.

[50] K. V. Srinivas, A. W. Eckford, and R. S. Adve, "Molecular communication in fluid media: The additive inverse gaussian noise channel," *IEEE Transactions on Information Theory*, 2012.

[51] B. Atakan and O. B. Akan, "Single and multiple-access channel capacity in molecular nanonetworks," in *International Conference on Nano-Networks*, Springer.

[52] H. Arjmandi, A. Gohari, M. N. Kenari, and F. Bateni, "Diffusion-based nanonetworking: A new modulation technique and performance analysis," *IEEE Communications Letters*,

[53] Y. Murin, N. Farsad, M. Chowdhury, and A. Goldsmith, "On time-slotted communication over molecular timing channels," *Submitted to Nanocom*, 2016.

[54] M. J. Moore, T. Suda, and K. Oiwa, "Molecular communication: Modeling noise effects on information rate," *IEEE transactions on nanobioscience*,

[55] L. Felicetti, M. Femminella, and G. Reali, "Smart antennas for diffusion-based molecular communications," in *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication*, ACM, 2015.

[56] C. T. Chou, "Maximum a-posteriori decoding for diffusion-based molecular communication using analog filters," *Nanotechnology, IEEE Transactions on*, 2015.

[57] E. Katz, *Biomolecular information processing: From logic systems to smart sensors and actuators*. John Wiley & Sons, 2013.

[58] M. S. Leeson and M. D. Higgins, "Forward error correction for molecular communications," *Nano Communication Networks*, vol. 3, no. 3, pp. 161–167, 2012.

[59] T. Furubayashi, T. Nakano, A. Eckford, and T. Yomo, "Reliable end-to-end molecular communication with packet replication and retransmission," in *Global Communications Conference (GLOBECOM), 2015 IEEE*.

[60] P. He, Y. Mao, Q. Liu, and K. Yang, "Improving reliability performance of diffusion-based molecular communication with adaptive threshold variation algorithm," *International Journal of Communication Systems*, 2016.

[61] P.-J. Shih, C.-h. Lee, and P.-C. Yeh, "Channel codes for mitigating intersymbol interference in diffusion-based molecular communications," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, IEEE, 2012, pp. 4228–4232.

[62] J. Engebrecht, K. Nealson, and M. Silverman, "Bacterial bioluminescence: Isolation and genetic analysis of functions from *Vibrio fischeri*," *Cell*, vol. 32, no. 3, pp. 773 –781, 1983.

[63] J. Sambrook, *Molecular Cloning: A Laboratory Manual, Third Edition (3 volume set)*, 3rd. Jan. 2001, ISBN: 0879695773.

[64] F. R. Blattner, G. Plunkett, Bloch, *et al.*, "The complete genome sequence of *Escherichia coli* k-12.," *Science*, 1997.

[65] J. B. Andersen, C. Sternberg, Poulsen, *et al.*, "New unstable variants of green fluorescent protein for studies of transient gene expression in bacteria.," *Applied and environmental microbiology*, vol. 64, 1998.

[66] J. R. van der Meer and S. Belkin, "Where microbiology meets microengineering: Design and applications of reporter bacteria," *Nature Reviews Microbiology*, 2010.

[67] R. D. Whitaker, S. Pember, B. C. Wallace, C. E. Brodley, and D. R. Walt, "Single cell time-resolved quorum responses reveal dependence on cell density and configuration," *Journal of Biological Chemistry*, vol. 286, 2011.

[68] A. Meyer, J. A. Megerle, C. Kuttler, J. Müller, C. Aguilar, L. Eberl, B. A. Hense, and J. O. Rädler, "Dynamics of AHL mediated quorum sensing under flow and non-flow conditions.," *Phys Biol*, vol. 9, no. 2, p. 026 007, 2012.

[69] S. Park, X. Hong, W. S. Choi, and T. Kim, "Microfabricated ratchet structure integrated concentrator arrays for synthetic bacterial cell-to-cell communication assays," *Lab Chip*, 2012.

[70] A. Prindle, P. Samayoa, I. Razinkov, *et al.*, "A sensing array of radically coupled genetic 'biopixels'.," *Nature*, 2012.

[71] A. Groisman, C. Lobo, H. Cho, J. K. Campbell, Y. S. Dufour, A. M. Stevens, and A. Levchenko, "A microfluidic chemostat for experiments with bacterial and yeast cells," *Nature methods*, vol. 2, no. 9, pp. 685–689, Sep. 2005.

[72] J. R. Anderson, D. T. Chiu, H. Wu, O. J. Schueller, and G. M. Whitesides, "Fabrication of microfluidic systems in poly(dimethylsiloxane)," *Electrophoresis*, 2000.

[73] V. Anantharam and S. Verdú, "Bits through queues," *IEEE Transactions on Information Theory*, 1996.

[74] T. M. Cover and J. A. Thomas, *Elements of Information Theory*.

[75] M. J. Moore and T. Nakano, "Addressing by beacon distances using molecular communication," *Nano Communication Networks*, 2011.

[76] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics-based biochips: Technology issues, implementation platforms, and design-automation challenges," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 25, no. 2, pp. 211–223, 2006.

[77] *Fabrication of microfluidic systems in poly (dimethylsiloxane) electrophoresis.*

[78] T. Bohman, "A sum packing problem of erdös and the conway-guy sequence," *Proceedings of the American Mathematical Society*,

[79] R. K. Guy, "Sets of integers whose subsets have distinct sums," *North-Holland Mathematics Studies*, 1982.

[80] G. E. Andrews, *The theory of partitions*, 2. Cambridge university press, 1998.

[81] R. M. Davis, R. Y. Muller, and K. A. Haynes, "Can the natural diversity of quorum-sensing advance synthetic biology?" *Synthetic Biology engineering complexity and refactoring cell capabilities*, 2015.

[82] Y. Jian, B. Krishnaswamy, C. M. Austin, A. O. Bicen, J. Perdomo, S. Patel, I. F. Akyildiz, C. Forest, and R. Sivakumar, "Nanons3:simulating bacterial molecular communication based nanonetworks in network simulator 3," in *3rd ACM International Conference on Nanoscale Computing and Communication 2016 (ACM NanoCom'16)*.

[83] E. Gul, B. Atakan, and O. B. Akan, "Nanons: A nanoscale network simulator framework for molecular communications," *Nano Communication Networks*, vol. 1, no. 2, pp. 138–156, 2010.

[84] I. Llatser, D. Demiray, A. Cabellos-Aparicio, D. T. Altilar, and E. Alarcón, "N3sim: Simulation framework for diffusion-based molecular communication nanonetworks," *Simulation Modelling Practice and Theory*, vol. 42, pp. 210–222, 2014.

[85] G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Nano-sim: Simulating electromagnetic-based nanonetworks in the network simulator 3," *6th International ICST Conference on Simulation Tools and Techniques*, pp. 203–210, 2013.

[86] (). Calcomsim: Https://sites.google.com/site/calcomsimulator/.

[87] (). Comsol-multiphysics: Https://www.comsol.com/comsol-multiphysics.

[88] L Felicetti, M Femminella, and G Reali, "A simulation tool for nanoscale biological networks," *Nano Communication Networks*, vol. 3, no. 1, pp. 2–18, 2012.

[89] A. Akkaya and T. Tugcu, "Dmcs: Distributed molecular communication simulator," in *8th ICST International Conference on Body Area Networks*, 2013, pp. 468–471.

[90] Y. Jian, B. Krishnaswamy, C. M. Austin, A. O. Bicen, J. E. Perdomo, S. C. Patel, I. F. Akyildiz, C. R. Forest, and R. Sivakumar, "Nanons3: Simulating bacterial molecular communication based nanonetworks in network simulator 3," *3rd ACM*

International Conference on Nanoscale Computing and Communication, pp. 17–23, 2016.

[91] L. Felicetti, M. Femminella, G. Reali, P. Gresele, M. Malvestiti, and J. N. Daigle, "Modeling cd40-based molecular communications in blood vessels," *IEEE Transactions on Nanobioscience*, vol. 13, no. 3, pp. 230–243, 2014.

[92] *Ns-3*, `https://www.nsnam.org/`.

[93] E. Weingartner, H. Vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," *IEEE International Conference on Communications*, pp. 1–5, 2009.

[94] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984, vol. 38.

[95] N. Farsad, N.-R. Kim, A. Eckford, and C.-B. Chae, "Channel and noise models for nonlinear molecular communication systems,"

[96] A. Einolghozati, M. Sardari, A. Beirami, and F. Fekri, "Capacity of discrete molecular diffusion channels," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, IEEE, 2011, pp. 723–727.

[97] M. Pierobon and I. F. Akyildiz, "Capacity of a diffusion-based molecular communication system with channel memory and molecular noise," *IEEE Transactions on Information Theory*, vol. 59, no. 2, pp. 942–954, 2013.

[98] B. Krishnaswamy and R. Sivakumar, "Source addressing and medium access control in bacterial communication networks," in *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication*, ACM, 2015.

[99] J. Bonnet, P. Subsoontorn, and D. Endy, "Rewritable digital data storage in live cells via engineered control of recombination directionality," *Proceedings of the National Academy of Sciences*, 2012.

[100] B. Wang, R. I. Kitney, N. Joly, and M. Buck, "Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology," *Nature communications*,

[101] *Nih grant:* `https://grants.nih.gov/grants/guide/pa-files/PAR-17-160.html`.

[102] *Nsf grant:* https://www.nsf.gov/awardsearch/showAward?AWD_ID=1810802&HistoricalAwards=false.

[103] *Nsf grant:* https://www.nsf.gov/awardsearch/showAward?AWD_ID=1707068&HistoricalAwards=false.

[104] *Farmview*, https://www.cmu.edu/work-that-matters/farmview/.

[105] C.-F. Shih, B. Krishnaswamy, and R. Sivakumar, "Rhythm: Achieving scheduled wifi using purely distributed contention in wlans," in *Global Communications Conference (GLOBECOM), 2015 IEEE*, IEEE, 2015, pp. 1–7.

[106] X. Liu, H. Yuk, S. Lin, G. A. Parada, T.-C. Tang, E. Tham, C. de la Fuente-Nunez, T. K. Lu, and X. Zhao, "3d printing of living responsive materials and devices," *Advanced Materials*, 2017.

[107] *Nsf grant:* https://www.nsf.gov/awardsearch/showAward?AWD_ID=1706620&HistoricalAwards=false.

[108] *Nih grant:* https://grants.nih.gov/grants/guide/rfa-files/RFA-ES-16-005.html.

[109] *Nsf grant:* https://nsf.gov/awardsearch/showAward?AWD_ID=1706994&HistoricalAwards=false.